

Integração e Avaliação de um Módulo Linux DCCP ao Simulador de Redes NS-3

Ivo Calado^{1,3}, Romeryto Lira², Leandro Sales², Hyggo Almeida², Angelo Perkusich¹

¹Departamento de Engenharia Elétrica –
Universidade Federal de Campina Grande (UFCG)
Cep: 58429-140 Campina Grande – PB – Brasil

²Departamento de Sistemas e Computação –
Universidade Federal de Campina Grande (UFCG)
Cep: 58429-140 Campina Grande – PB – Brasil

³Doutorando do Programa de Pós-Graduação em Engenharia Elétrica da UFCG
{ivocalado, romeryto, leandro, hyggo, perkusich}@embedded.ufcg.edu.br

Resumo. *A integração de pilhas de protocolos reais em simuladores de redes representa uma poderosa ferramenta para realizar simulações de redes. No entanto, todas as abordagens existentes apresentam enfoque no protocolo TCP. Neste trabalho propõe-se a integração do DCCP (Datagram Congestion Control Protocol) Linux ao simulador de redes NS-3, possibilitando a realização de simulações de redes que façam uso do protocolo DCCP. Os resultados obtidos mostram que o protocolo DCCP executado no simulador consegue obter altos índices de utilização de canal e Equidade Inter e Intra-protocolo.*

Abstract. *The integration of real network protocol stacks in network simulators represents a powerful tool for network simulations. However, all existing approaches have focused only on the TCP protocol. This paper proposes the integration of the Linux DCCP (Datagram Congestion Control Protocol) into the network simulator NS-3, enabling the construction of network simulations involving DCCP protocol. The results show that DCCP achieves levels of channel utilization, as well as intra and inter protocol fairness.*

1. Introdução

Na área de redes de computadores, os protocolos de rede são considerados peças fundamentais em qualquer tipo de comunicação, estando associados tanto ao conteúdo transmitido quanto ao meio por onde os dados são transmitidos. Deste modo, é notável o contínuo interesse na realização de pesquisas que envolvam o desenvolvimento de novos protocolos e melhorias dos protocolos atualmente padronizados, principalmente por parte da comunidade acadêmica e da indústria.

Do conjunto de ferramentas utilizadas durante o desenvolvimento de aplicações e análise comportamental dos protocolos de rede encontram-se os simuladores de rede. Um simulador de rede é um software que modela transmissões de dados de maneira semelhante ao que acontece em redes reais. Para tal, é criado um modelo do sistema ou protocolo de rede a ser simulado, de modo que suas propriedades possam ser alteradas e o seu comportamento analisado. Entre as vantagens da utilização de simuladores de rede encontram-se: (1) simplicidade na implementação do protocolo em comparação à implementação real, (2) reproducibilidade de cenários e resultados e (3) escalabilidade nos cenários de estudo, entre outros [Breslau et al. 2000].

A validade dos resultados das simulações está intimamente ligada aos modelos dos protocolos utilizados como base na execução do protocolo a ser avaliado. Se os comportamentos dos modelos estiverem próximos dos comportamentos dos protocolos

reais, então os resultados de desempenho obtidos a partir das simulações tendem a convergir para valores bastante próximos do que poderia ser encontrado na execução em um cenário real. Contudo, em virtude dos constantes aprimoramentos e otimizações aplicados às implementações dos sistemas operacionais, os modelos existentes nos simuladores de redes tendem a se tornar obsoletos rapidamente.

Com base em tal problemática, diversos grupos de pesquisas [Zec 2003, Ely et al. 2001, Huang et al. 1999, Jansen and McGregor 2005] têm trabalhado em propostas que possibilitem a utilização em simuladores de rede de pilhas de protocolos implementadas em sistemas operacionais. A partir de tal abordagem, consegue-se minimizar os efeitos das deficiências apontadas tanto na validação através de experimentos em redes reais quanto em estudos com base em simulações de rede.

Todavia, as propostas encontradas na literatura tendem a oferecer suporte ao protocolo TCP [RFC 1981]. Contudo, embora o protocolo TCP seja amplamente utilizado em uma vasta gama de domínios, no domínio das aplicações multimídia prefere-se utilizar outros protocolos cujas características possibilitam melhor desempenho de transmissão, como o protocolo DCCP (*Datagram Congestion Control Protocol*) [Kohler et al. 2006].

Neste artigo, apresenta-se o processo e os resultados obtidos sobre a integração do protocolo DCCP disponível no núcleo do Linux ao simulador de redes NS-3 (Network Simulator 3) [NSNam 2011]. Para tal, foi utilizado como base o arcabouço *Network Simulation Cradle* [Jansen and McGregor 2005], objetivando-se a comunicação entre o referido simulador de redes e o código do protocolo DCCP.

Com o intuito de validar o trabalho desenvolvido, foram realizados experimentos a fim de analisar o protocolo DCCP executado no simulador sob os seguintes critérios: utilização do canal em função da largura de banda, equidade intra-protocolo e equidade interprotocolo. De acordo com os resultados, constatou-se que o módulo do protocolo DCCP implementado no simulador apresenta altos níveis de utilização do canal, equidade entre conexões DCCP e equidade com conexões TCP para os tratamentos definidos.

Este trabalho está organizado como segue. Na Seção 2 discutem-se os principais aspectos do protocolo DCCP. Na seção seguinte são apresentados alguns trabalhos relacionados à presente pesquisa. Na Seção 4 discutem-se os detalhes do processo de integração NS-3/DCCP, ao passo que na Seção 5 apresenta-se a metodologia para os estudos de avaliação. Na Seção 6 apresentam-se os resultados obtidos e, por fim, na Seção 7 apresentam-se as considerações finais deste trabalho.

2. O protocolo DCCP

O protocolo DCCP (do inglês *Datagram Congestion Control Protocol*), definido sob a RFC 4340 [Kohler et al. 2006] é um protocolo da camada de transporte que provê comunicação *unicast* entre dois *hosts*, controle de congestionamento e entrega não confiável de pacotes. O DCCP é projetado para aplicações que desejam proporcionar um controle de congestionamento, visando a não degradação dos recursos da rede, assim como ocorre com o TCP, mas sem necessitar da garantia de entrega dos dados, semelhante ao UDP.

Deste modo, o objetivo do DCCP é prover uma forma padronizada de implementar mecanismos de controle de congestionamento e negociação, voltado para aplicações de tempo real. O DCCP realiza estabelecimento, finalização e negociação de propriedades da conexão de maneira confiável e realiza a confirmação dos pacotes recebidos dando suporte a pacotes marcados no campo ECN.

Cada conexão DCCP estabelecida é bidirecional, isto é, é possível o tráfego tanto de pacotes de dados quanto de confirmação de entrega de pacotes (ACKs) em ambas as direções. No entanto, logicamente, é como se existissem sub-fluxos (mais conhecidos como *half-connection*), visto que, em cada direção da transmissão, são passados tanto dados da aplicação quanto de confirmação de pacotes. Ademais, cada sub-fluxo tem a

possibilidade de definir o algoritmo de controle de congestionamento que irá utilizar na transmissão. Ou seja, dois diferentes algoritmos de controle de congestionamento podem ser utilizados na mesma conexão em direções diferentes. Contudo, apesar de virtualmente isolados, os dois sub-fluxos compartilham o mesmo canal de transmissão. Por exemplo, pacotes de confirmação podem ser utilizados para, além de confirmar pacotes recebidos, fazerem envio de novos pacotes de dados, utilizando a técnica conhecida como “*piggy-back*” [Kurose and Ross 2006, Kohler et al. 2006].

Uma importante característica presente no DCCP é a possibilidade de definir qual mecanismo de controle de congestionamento irá gerenciar o envio de dados em cada subfluxo da transmissão, podendo ser definido tanto antes quanto durante a transmissão. O principal objetivo deste mecanismo é permitir que fluxos com características diferentes possam utilizar o controle de congestionamento mais adequado. Para possibilitar o acoplamento e a utilização dos mecanismos de controle de congestionamento na abordagem supracitada, uma estrutura modular foi projetada de forma a separar o núcleo do protocolo dos mecanismos de controle de congestionamento presentes, possibilitando também que novos mecanismos possam ser adicionados facilmente ao DCCP.

Para permitir selecionar um controle de congestionamento entre diferentes mecanismos, cada mecanismo é referenciado por um identificador, conhecido como CCID (*Congestion Control Identifiers*) [Kohler et al. 2006]. A atribuição destes identificadores é realizada pela IANA (*Internet Assigned Numbers Authority*), onde valores nas faixas de 0 a 1 e 5 a 247 encontram-se disponíveis para utilização. No presente momento, encontram-se padronizados três CCIDs: CCID-2 [Floyd and Kohler 2006], CCID-3 [Floyd et al. 2006] e CCID-4 [Floyd and Kohler 2009].

No que concerne a sua disponibilização, atualmente o DCCP está implementado nos sistemas operacionais FreeBSD e Linux.

3. Trabalhos Relacionados

3.1. *Implementation and Evaluation of a DCCP Module for Network Simulator 2*

Perez e Ocampo [Perez and Ocampo 2009] apresentam um módulo do protocolo DCCP voltado para o simulador de redes NS-2. Em tal abordagem, os elementos do protocolo são modelados como agentes seguindo a arquitetura do referido simulador. São providas implementações para os algoritmos de controle de congestionamento CCID-2 [Floyd and Kohler 2006] e CCID-3 [Floyd et al. 2006]. Com o intuito de avaliar a implementação realizada, os autores realizaram uma série de experimentos de modo a comparar o módulo do DCCP para NS-2 em relação a dois aspectos. O primeiro está relacionado à implementação do protocolo TCP existente no NS-2, ao passo que o segundo está relacionado ao comportamento do protocolo DCCP em uma rede real. Em ambos os experimentos realizados, o módulo conseguiu obter níveis satisfatórios de consumo de largura de banda, considerando-se os níveis esperados de desempenho. Porém, os autores não apresentam uma análise estatística mais detalhada quanto ao comportamento do DCCP em comparação com cenários reais, o que tornaria possível obter conclusões mais consistentes sobre o trabalho.

3.2. *Measured Comparative Performance of TCP Stacks*

Jansen e McGregor [Jansen and McGregor 2006] apresentam um estudo comparativo das diferentes implementações do protocolo TCP, cada uma destas pertencente a um sistema operacional distinto. Para cada sistema operacional foram coletados dados empíricos baseados em um experimento composto de 100 ensaios. Baseados nestes dados foram obtidos dados relativos ao desempenho deste protocolo para cada sistema operacional, verificando-se se havia diferenças significativas de desempenho nas diferentes implementações do protocolo TCP.

Em seguida, descreve-se brevemente o cenário para os ensaios e demonstra-se que há diferenças significativas de desempenho entre as distintas implementações do protocolo TCP. Porém, no referido trabalho não apresentam-se resultados acerca do impacto de possíveis más implementações do TCP e as discrepâncias de desempenho observada ao utilizar cada uma delas.

3.3. Network Simulation Cradle

O *Network Simulation Cradle* é um arcabouço que possibilita a integração em simuladores de rede de pilhas de protocolos implementadas em sistemas operacionais [Jansen and McGregor 2005]. Ele foi projetado de forma a diminuir ao máximo as alterações necessárias em um protocolo da camada de transporte que venha a ser plugado no arcabouço.

Na Figura 1 ilustra-se como o simulador de rede se integra ao arcabouço NSC. O NSC se comunica com o simulador de redes de duas maneiras, simulando a entrada de dados na pilha de código real e a saída. Para o primeiro caso, o NSC serve como um adaptador modificando as mensagens provenientes do simulador de redes com o intuito de fazer o envio na pilha de protocolos. Por sua vez, ele também realiza uma abstração da camada de *enlace* permitindo a reinserção dos pacotes a serem enviados pelo simulador de redes.

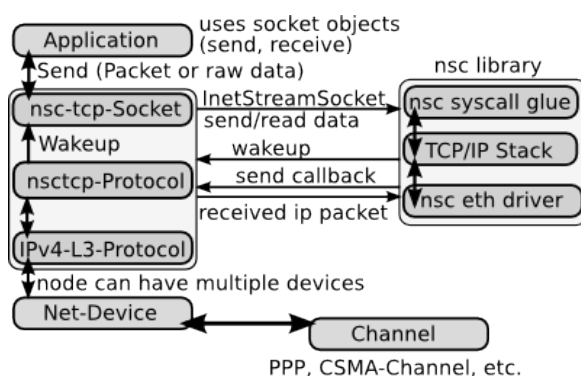


Figura 1. Fluxo de comunicação entre o NS-3 e o *Network Simulation Cradle* [Jansen and McGregor 2008].

4. Integração do DCCP Linux ao Simulador de Redes NS-3

O processo de integração do protocolo DCCP ao simulador de redes NS-3 foi realizado em três etapas, descritas a seguir:

1. **Estudo da estrutura do NSC tendo como base o TCP:** A primeira etapa do desenvolvimento do módulo pretendido consistiu da realização de um estudo a respeito do funcionamento interno do arcabouço NSC e de como este se relaciona com o simulador de redes NS-3. Para tal, utilizou-se como base o protocolo da camada de transporte TCP. Foram analisadas as interfaces de comunicação. A partir de tal etapa tornou-se possível identificar os passos subsequentes na implementação.
2. **Implementação da camada de acesso no NS-3 e NSC:** Esta etapa representou a implementação propriamente dita, sendo subdividida em duas etapas. A primeira teve como objetivo a implementação do protocolo no arcabouço NSC. Para tal, foram implementadas as funções e rotinas necessárias para o estabelecimento de conexões, transmissão de dados e, por fim, fechamento da conexão. O segundo momento da implementação foi dedicado à implementação do modelo básico do protocolo DCCP no simulador de redes NS-3. Por se tratar de um

simulador de redes ainda em desenvolvimento, alguns protocolos ainda não se encontram implementados, entre eles, o protocolo DCCP. Neste sentido, o objetivo desta pesquisa foi como levar a implementação do protocolo DCCP presente no núcleo do sistema operacional Linux para que pudesse ser integrado ao NS-3.

3. **Backport da versão de teste do protocolo DCCP para a versão 2.6.29:** O arcabouço NSC está na versão 0.52 e oferece suporte às seguintes pilhas de protocolos:

- Linux 2.6;
- Linux 2.6.18;
- Linux 2.6.26;
- Linux 2.6.29;
- Lwip-1.3.0;
- OpenBSD 3.0.

Entre as versões suportadas do sistema Linux, todas são baseadas na versão básica do núcleo do Linux, a qual é considerada a versão mais estável, porém sem todas as otimizações e modificações disponibilizadas recentemente pelos desenvolvedores do Linux. Dito isto, decidiu-se utilizar a última versão do protocolo DCCP, a qual ainda não foi totalmente integrada à versão básica do Linux. Para isto, existiam duas opções:

- tornar o NSC compatível com a última versão do Linux;
- migrar a última implementação do DCCP, disponível no Linux 3.0, para a última versão do NSC, que suporta o Linux 2.6.29.

Em virtude da complexidade envolvida em realizar a migração da versão completa do núcleo do Linux, optou-se por realizar a migração apenas da implementação do DCCP para a versão 2.6.29. O principal desafio decorrente dessa escolha foi resolver as inconsistências das assinaturas de funções e definições das estruturas de dados entre a versão do Linux 3.0 e a Linux 2.6.29.

No presente momento, o projeto encontra-se em estágio experimental, estando disponível para download através do link <https://github.com/txithihausen/nsc-dccp>.

5. Metodologia de avaliação

O processo de avaliação do trabalho desenvolvido consistiu na análise do comportamento do protocolo DCCP sob diferentes cenários de simulação no NS-3. De maneira específica, o protocolo DCCP foi avaliado ao se utilizar os dois principais mecanismos de controle de congestionamento, o CCID-2 [Floyd and Kohler 2006] e o CCID-3 [Floyd et al. 2006].

5.1. Métricas avaliadas

Foram consideradas algumas das métricas especificadas no documento *Metrics for the Evaluation of Congestion Control Mechanisms* [Floyd 2008]. Neste documento apresentam-se as recomendações da IETF¹ sobre a escolha de métricas para avaliar o desempenho de algoritmos para controle de congestionamento. Neste sentido, considerou-se as seguintes métricas: *Vazão*, *Equidade Intra-protocolo* e *Equidade*.

5.1.1. Vazão

A vazão é a principal métrica estudada pela maioria dos trabalhos disponíveis na literatura. A partir da avaliação da vazão torna-se possível obter uma visão geral do nível de utilização do canal de transmissão. A vazão pode ser analisada tomando como base tanto a utilização do canal por apenas uma transmissão do protocolo bem como através da análise da vazão agregada de um conjunto de fluxos de dados do referido protocolo.

¹ Acrônimo de *Internet Engineering Task Force*.

Considerando a utilização da primeira abordagem, torna-se possível a realização de uma análise a respeito do desempenho dos algoritmos de controle de congestionamento estudados. Por outro lado, ao realizar uma análise de utilização do canal a partir de um grande conjunto de fluxos de dados, torna-se possível analisar a estabilidade do algoritmo de controle de congestionamento diante de um ambiente de execução com diversos fluxos de dados.

No contexto da avaliação realizada, considerou-se o nível de utilização do canal de transmissão em função do número de conexões existentes. O objetivo principal dessa escolha foi detectar se o protocolo iria manter a estabilidade com um nível alto de utilização do canal, variando-se a quantidade de conexões estabelecidas.

5.1.2. Equidade

O critério de Equidade ou *fairness* possibilita a realização de uma avaliação da capacidade do protocolo em dividir o canal de transmissão compartilhado em condições semelhantes de transmissão. Em um sistema considerado completamente equânime, o recurso compartilhado é dividido de maneira igualitária. Em outras palavras, se um canal de transmissão possui largura de banda X e existem Z conexões estabelecidas, a taxa de transmissão de cada uma das conexões após o período de convergência do algoritmo de controle de congestionamento será de $\frac{X}{Z}$.

De modo a avaliar o nível de equidade apresentado pelo conjunto de transmissões estudadas, considerou-se o índice Jain [Jain et al. 1984] (*Jain Fairness index*). Este índice é definido de acordo com a Equação 1:

$$idx_{jain} = \frac{(\sum_{i=1}^n x_i)^2}{n * \sum_{i=1}^n (x_i^2)} \quad (1)$$

onde:

- n define o número de conexões;
- x_i define a vazão de uma determinada transmissão.

O índice varia entre 0 (zero) e 1, onde um valor próximo a 1 indica um sistema equânime enquanto que o valor próximo a 0 (zero) indica um sistema não equânime.

5.2. Equidade *intra* e *inter* protocolos

Um outro aspecto a ser analisado é a equidade *intra* e *inter* protocolo. O primeiro caso visa analisar como uma determinada transmissão se comporta em relação a outra transmissão do mesmo protocolo. Já o segundo caso especifica o nível de equidade de uma transmissão em relação a uma transmissão pertencente a outro protocolo. Deste modo, foram realizadas análises de equidade *intra*-protocolo e *inter*-protocolo, utilizando o TCP como protocolo de comparação.

5.3. Descrição dos experimentos

Com base na definição das métricas a serem estudadas foram definidos três grupos de experimentos. Cada um dos experimentos teve por objetivo demonstrar o comportamento do protocolo DCCP a partir de uma determinada métrica. A seguir serão descritas em detalhes as características comuns a todos os experimentos e em seguida serão apresentadas as características específicas.

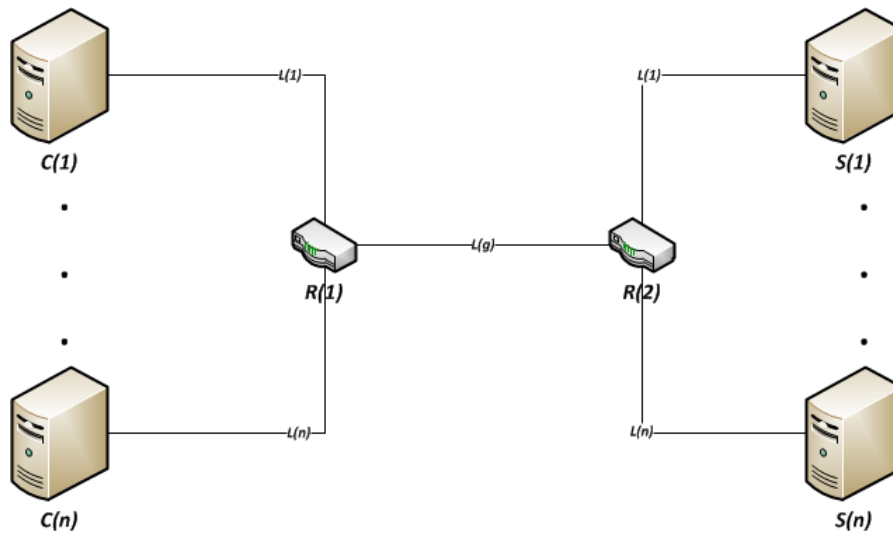


Figura 2. Topologia *Dumbbell* utilizada nas simulações.

5.3.1. Definição da População

A população é constituída por dados provenientes da variável dependente *vazão* obtidos a partir da execução dos algoritmos de controle de congestionamento CCID-2 e CCID-3 do DCCP. Os dados foram coletados a partir de simulações realizadas pelo NS-3, onde cada ensaio foi executado durante o tempo de 100 segundos. Uma amostra da vazão foi coletada a cada segundo, totalizando 100 amostras.

5.3.2. Definição da topologia de rede

A topologia usada para os experimentos foi a *Dumbbell*, útil para representar uma rede com canal de transmissão saturado. Na Figura 2 ilustra-se a representação gráfica desta topologia.

Para esta topologia, tem-se que $C = \{c_1, c_2, \dots, c_{i-1}, n_i\}$ representa o conjunto de nós transmissores, $S = \{s_1, s_2, \dots, s_{i-1}, s_i\}$ representa o conjunto de nós receptores, $R = \{r_1, r_2\}$ representa o conjunto de roteadores do canal saturado e $L = \{l_1, l_2, \dots, l_{i-1}, l_i\}$ representa o conjunto de *enlaces* ligando C a r_1 e S a r_2 , sendo i o índice de cada nó em R . Ademais, l_g indica o *enlace* saturado conectando r_1 e r_2 . Em cada experimento, realizou-se uma transmissão envolvendo os nós c_i e s_i .

5.3.3. Definição de Variáveis

As variáveis envolvidas no experimento foram divididas em dois tipos: variáveis independentes e variáveis dependentes. A seguir são especificadas tais variáveis:

1. Variáveis independentes:

- nós: $c_1, c_2, \dots, c_{i-1}, c_i, s_1, s_2, \dots, s_{i-1}, s_i$;
- roteadores: r_1 e r_2 ;
- tamanho dos pacotes: 1400 bytes;
- largura de banda nos *enlaces* $l_1, l_2, \dots, l_{i-1}, l_i$: 1000 Mbps;
- atraso nos *enlaces* $l_1, l_2, \dots, l_{i-1}, l_i$: 1 ms;
- tipo de descarte de pacote nos roteadores: *Drop Tail*;
- tamanho da fila de descarte em r_1 e r_2 : 100 pacotes;

- taxa de transmissão da aplicação: 50 Mbps;
 - atraso em l_g : 15 ms;
2. Variáveis dependentes
 - vazão recebida.
 3. Fatores:
 - dependente de cada experimento (descritos a seguir).

Com base nos valores das variáveis independentes, o ambiente de experimento foi ajustado de modo a obter os dados de saída que são relativos à variável dependente estudada. Para isto também foram definidos os fatores, que são variáveis independentes que variam de um tratamento de experimento para outro.

De modo a obter um nível satisfatório de confiança, para cada tratamento foram realizadas 50 repetições de onde foram extraídas as médias da métrica *vazão* e os intervalos de confiança com um nível de confiança de 95%.

5.3.4. Experimento 1: Avaliar a utilização do canal em função do número de conexões DCCP

Este experimento teve por objetivo avaliar o protocolo DCCP em um ambiente onde a largura de banda do canal saturado foi de *10 Mbps*, a partir da avaliação dos níveis de utilização de canal em função do número de conexões. Para tal, foram definidos quatro tratamentos, onde em cada tratamento foi utilizado um número específico de conexões, conforme descritos abaixo:

- **Tratamento 1:** 1 conexão;
- **Tratamento 2:** 5 conexões;
- **Tratamento 3:** 10 conexões;
- **Tratamento 4:** 20 conexões.

Com base nos tratamentos definidos, possibilitou-se um estudo sobre o comportamento do DCCP tanto em um ambiente onde diversas conexões compartilham o canal quanto em um ambiente onde apenas uma transmissão tem a possibilidade de utilizar todos os recursos disponíveis.

Com o intuito de evitar a sincronia das transmissões, as transmissões foram configuradas para terem seu início entre 1 e 10 segundos iniciais da simulação. Além disso, a duração das transmissões segue a distribuição estatística Pareto [Jain 1991], tendo como valor médio da transmissão 100 segundos e limite máximo de 300 segundos. Tal escolha deve-se ao fato desta distribuição representar tipicamente as características das transmissões em um ambiente real, onde um conjunto reduzido de transmissões são responsáveis por grande parte da utilização dos recursos disponíveis.

5.3.5. Experimento 2: Avaliar a Equidade Intra-protocolo em função da largura de banda

Neste segundo experimento, objetivou-se analisar a equidade intra-protocolo apresentada pelo protocolo DCCP em relação a outras transmissões DCCP em redes com diferentes valores de largura de banda disponível. Neste contexto, foram estabelecidas 4 conexões DCCP com duração de 100 segundos cada em cenários com diferentes valores para largura de banda para o link l_g : 1 Mbps, 10 Mbps e 20 Mbps (Tratamentos 1, 2 e 3, respectivamente).

Ademais, visando proporcionar um cenário mais realístico nas transmissões, foram adicionados fluxos de segundo plano (*background*). Tais fluxos são caracterizados por

Tratamentos	CCID	
	CCID-2	CCID-3
Tratamento 1	9.434714 (9.402473 - 9.466954)	9.072729 (8.978206 - 9.167252)
Tratamento 2	9.561038 (9.542676 - 9.579399)	9.437969 (9.399554 - 9.476383)
Tratamento 3	9.580881 (9.567690 - 9.594072)	9.547308 (9.521554 - 9.573062)
Tratamento 4	9.605070 (9.596540 - 9.613600)	9.631690 (9.618673 - 9.644707)

Tabela 1. Experimento 1: Utilização de canal em função do número de conexões.

quatro transmissões TCP Reno, cuja taxa de transmissão máxima somada foi limitada a 20% da largura de banda no tratamento estudado. Por exemplo, no *Tratamento 3* foi definido um fluxo em *background* com taxa de transmissão máxima de 4 Mbps equivalente a 20% da taxa nominal do *enlace* l_q . De maneira semelhante ao experimento anterior, o início das transmissões de *background* foi definido de maneira aleatória seguindo uma distribuição uniforme com valores no intervalo entre 1 e 10 segundos.

5.3.6. Experimento 3: Avaliar a Equidade Inter-protocolo em função da largura de banda

Este experimento foi bastante semelhante ao experimento anterior. A diferença foi que utilizou-se duas transmissões DCCP e duas conexões TCP em vez de 4 transmissões DCCP. Deste modo tornou-se possível analisar como o protocolo DCCP se comporta na presença do protocolo TCP Reno.

6. Resultados e discussões

A partir dos dados coletados seguindo a metodologia apresentada na Seção 5, realizou-se a análise estatística para verificar o comportamento do protocolo DCCP. A seguir são apresentados os resultados e discussões obtidos.

6.1. Experimento 1

Na Tabela 1 são apresentados os resultados de ambos CCIDs estudados para **1, 5, 10 e 20** conexões (tratamentos 1, 2, 3 e 4, respectivamente), bem como os respectivos intervalos de confiança.

Para todos os cenários analisados, ambos os mecanismos de controle de congestionamento conseguiram atingir altos índices de utilização de canal durante as simulações. Com exceção do *Tratamento 4*, onde utilizou-se 20 conexões paralelas. Além disso, constatou-se que em todos os cenários estudados o desempenho do mecanismo CCID-2 foi superior ao do CCID-3.

Nas Figuras 3(a) e 3(b) ilustra-se a variação na utilização do canal em função do número de conexões ativas. Como pode-se observar, em todos os cenários o índice de utilização do canal foi superior a 90 %.

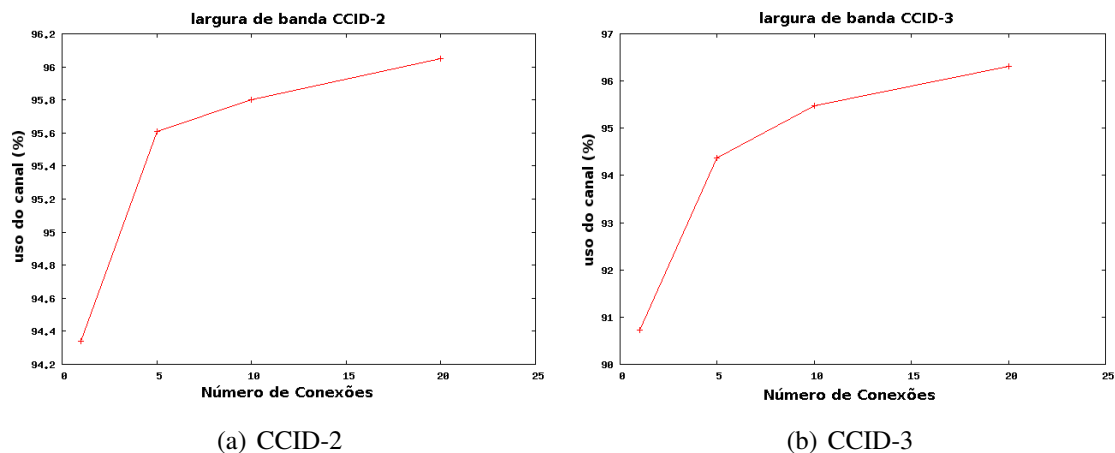


Figura 3. Utilização do canal em função do número de conexões.

Fluxos	Tratamentos		
	Tratamento 1	Tratamento 2	Tratamento 3
Fluxo 1	0.207085 (0.200854 - 0.213315)	2.117853 (2.077992 - 2.157715)	4.381183 (4.312049 - 4.450317)
Fluxo 2	0.205109 (0.198442 - 0.211775)	2.048496 (2.013230 - 2.083761)	4.221771 (4.151151 - 4.292391)
Fluxo 3	0.197544 (0.192004 - 0.203083)	2.117501 (2.079517 - 2.155484)	4.235960 (4.150374 - 4.321545)
Fluxo 4	0.198956 (0.192871 - 0.205042)	2.058682 (2.021975 - 2.095389)	4.209314 (4.147487 - 4.271141)

Tabela 2. Experimento 2: Vazão apresentada pelos fluxos utilizando o CCID-2 em Mbps

6.2. Experimento 2

Nas Tabelas 2 e 3 são apresentados os valores de vazão média obtidos pelos CCIDs 2 e 3, respectivamente. Além disso, são apresentados os valores dos intervalos de confiança com 95% de acurácia.

Como pode-se observar, há uma proximidade bastante elevada entre os valores apresentados pelos quatro fluxos. Tal proximidade é inversamente proporcional à largura de banda disponível no canal. Além disso, é possível verificar que o mecanismo de controle de congestionamento CCID-2 consegue perfazer a divisão do canal de maneira mais equânime que o CCID-3 e que ambos CCIDs conseguem fazer uso de quase toda largura de banda excedente. Por exemplo, os fluxos agregados do CCID-2 usam aproximadamente 80%, 84% e 85% da largura de banda disponível para os tratamentos 1, 2 e 3, respectivamente. Para o CCID-3, esses valores são respectivamente 77%, 78% e 75%.

Com base nos dados obtidos calculou-se o nível de equidade intra-protocolo apresentado, seguindo a Equação 1. Os resultados para ambos CCIDs são apresentados graficamente nas Figuras 4(a) e 4(b). Pode-se concluir que os dois CCIDs estudados apresentam altos níveis de equidade intra-protocolo apresentando valores para o índice Jain superiores a 0,9.

Fluxos	Tratamentos		
	Tratamento 1	Tratamento 2	Tratamento 3
Fluxo 1	0.174221 (0.169874 - 0.178568)	2.539716 (2.502645 - 2.576788)	6.329122 (6.228070 - 6.430174)
Fluxo 2	0.209642 (0.203656 - 0.215628)	2.010331 (1.979912 - 2.040751)	3.746072 (3.696705 - 3.795439)
Fluxo 3	0.179670 (0.175228 - 0.184113)	1.712802 (1.683754 - 1.741849)	2.974325 (2.934512 - 3.014138)
Fluxo 4	0.208265 (0.202990 - 0.213539)	1.610884 (1.587355 - 1.634412)	2.588482 (2.548204 - 2.628759)

Tabela 3. Experimento 2: Vazão apresentada pelos fluxos utilizando o CCID-3 em Mbps

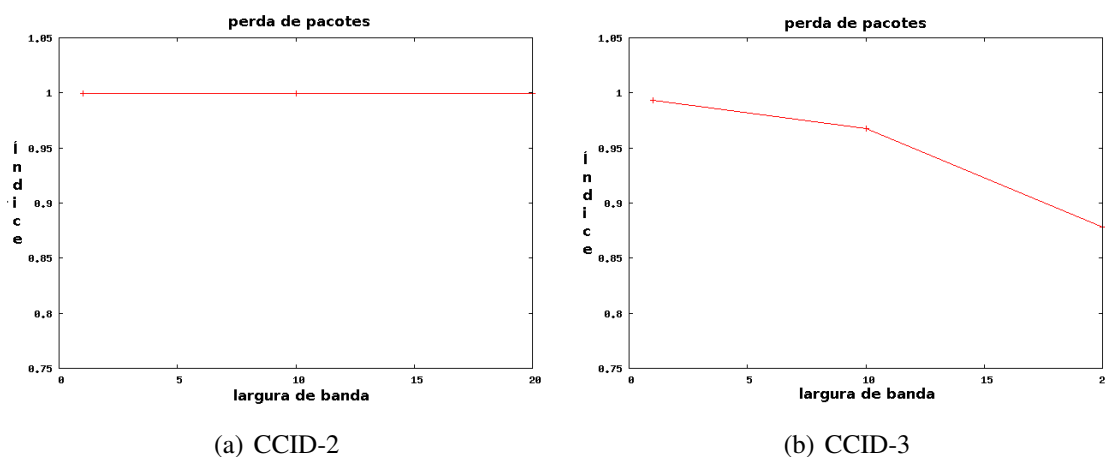


Figura 4. Índice de Equidade Intra-protocolo.

6.3. Experimento 3

De maneira análoga ao experimento anterior, são apresentados nas Tabelas 4 e 5 os valores médios, acrescidos do intervalo de confiança para cada mecanismo de controle de congestionamento.

Observa-se que o *Fluxo 1* e *2* representam transmissões DCCP enquanto os outros dois representam transmissões TCP Reno. Nas Figuras 5(a) e 5(b) apresenta-se a evolução dos valores de índice de equidade inter-protocolo de acordo com o uso do CCID-2 e CCID-3.

Em decorrência dos resultados apresentados, pode-se concluir que para ambos os CCIDs estudados, o nível de equidade inter-protocolo é menor que o de equidade intra-protocolo. De acordo com os valores dos índices Jain aliados aos resultados de vazão obtidos, constatou-se que os fluxos TCP Reno são menos agressivos que os fluxos DCCP. Esta diferença de desempenho é reconhecida e já havia sido reportada em outro trabalho [Lai 2008].

7. Considerações Finais

Os simuladores de redes representam uma importante ferramenta utilizada no processo de desenvolvimento, testes, depuração e estudo de protocolos de comunicação. Entre os

Fluxos	Tratamentos		
	Tratamento 1	Tratamento 2	Tratamento 3
Fluxo 1 (CCID-2)	0.302784 (0.297509 - 0.308059)	3.114942 (3.077384 - 3.152500)	6.248485 (6.162116 - 6.334855)
Fluxo 2 (CCID-2)	0.300577 (0.294191 - 0.306963)	3.099251 (3.063351 - 3.135150)	6.407751 (6.332483 - 6.483018)
Fluxo 3 (TCP Reno)	0.087097 (0.083035 - 0.091158)	0.738976 (0.714774 - 0.763178)	1.394663 (1.365055 - 1.424272)
Fluxo 4 (TCP Reno)	0.087653 (0.083625 - 0.091681)	0.701696 (0.681055 - 0.722337)	1.321347 (1.286970 - 1.355724)

Tabela 4. Experimento 3: Vazão apresentada pelos fluxos utilizando o CCID-2 e TCP Reno em Mbps

Fluxos	Tratamentos		
	Tratamento 1	Tratamento 2	Tratamento 3
Fluxo 1 (CCID-3)	0.271129 (0.265143 - 0.277115)	3.169988 (3.128600 - 3.211376)	6.946985 (6.862293 - 7.031677)
Fluxo 2 (CCID-3)	0.252563 (0.246339 - 0.258788)	2.554909 (2.521560 - 2.588257)	4.698422 (4.634636 - 4.762209)
Fluxo 3 (TCP Reno)	0.124724 (0.120135 - 0.129312)	0.910817 (0.886842 - 0.934793)	1.715388 (1.658278 - 1.772498)
Fluxo 4 (TCP Reno)	0.124301 (0.119533 - 0.129069)	0.895554 (0.872256 - 0.918853)	1.719668 (1.670824 - 1.768512)

Tabela 5. Experimento 3: Vazão apresentada pelos fluxos utilizando o CCID-3 e TCP Reno em Mbps

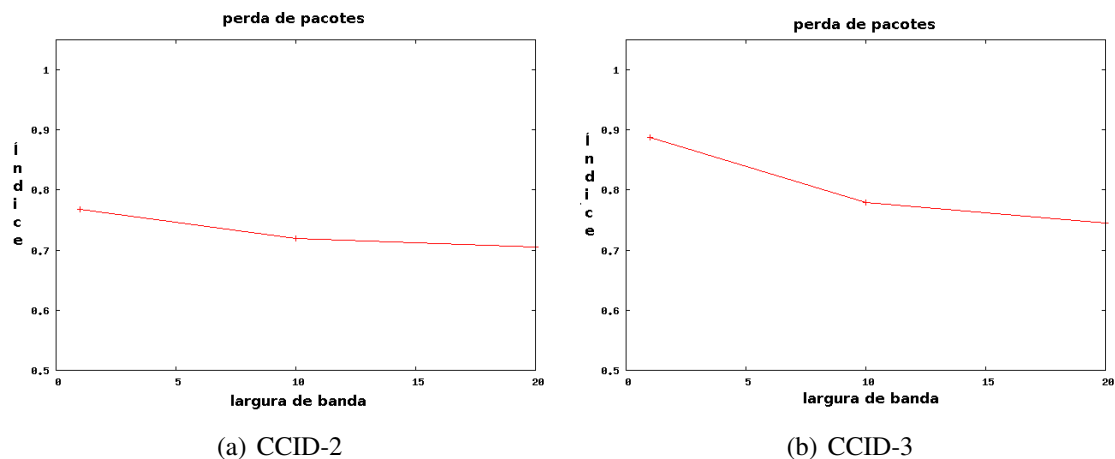


Figura 5. Índice de Equidade Inter-protocolo.

diversos simuladores de rede existentes, o NS-3 vem se destacando por oferecer uma nova opção em substituição ao já consagrado simulador de redes NS-2. No entanto, o suporte a diversos protocolos ainda não é oferecido no NS-3.

Neste trabalho apresentou-se o processo de integração da implementação do protocolo DCCP em Linux ao simulador de redes NS-3. A partir disso, torna-se possível a realização de simulações envolvendo esse importante protocolo da camada de transporte sem a complexidade de se configurar redes reais para a realização de diferentes estudos.

A partir dos experimentos realizados constatou-se que os algoritmos de controle de congestionamento disponibilizados no protocolo DCCP conseguem obter bom desempenho a partir da análise dos níveis de utilização do canal e de equidade intra e inter-protocolo.

Como trabalhos futuros sugere-se a avaliação do uso do DCCP em transmissões de dados multimídia, assim como a reprodução dos resultados obtidos em outros trabalhos que utilizaram redes reais utilizando o NS-3 com o DCCP integrado.

Referências

- [RFC 1981] (1981). TCP: Transmission Control Protocol. <http://tools.ietf.org/html/rfc793>. Último acesso em Novembro 2011.
- [Breslau et al. 2000] Breslau, L., Estrin, D., Fall, K., Floyd, S., Heidemann, J., Helmy, A., Huang, P., McCanne, S., Varadhan, K., Xu, Y., and Yu, H. (2000). Advances in network simulation. *Computer*, 33(5):59–67.
- [Ely et al. 2001] Ely, D., Savage, S., and Wetherall, D. (2001). Alpine: a user-level infrastructure for network protocol development. In *USITS'01: Proceedings of the 3rd conference on USENIX Symposium on Internet Technologies and Systems*, page 15, Berkeley, CA, USA. USENIX Association.
- [Floyd 2008] Floyd, S. (2008). Metrics for the Evaluation of Congestion Control Mechanisms. <http://www.faqs.org/rfcs/rfc5166.html>. Último acesso em Novembro 2011.
- [Floyd and Kohler 2006] Floyd, S. and Kohler, E. (2006). Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like Congestion Control. <http://www.ietf.org/rfc/rfc4341.txt>. Último acesso em Novembro 2011.

- [Floyd and Kohler 2009] Floyd, S. and Kohler, E. (2009). Profile for Datagram Congestion Control Protocol (DCCP) Congestion ID 4: TCP-Friendly Rate Control for Small Packets (TFRC-SP). <http://tools.ietf.org/html/rfc5622>. Último acesso Novembro 2011.
- [Floyd et al. 2006] Floyd, S., Kohler, E., and Padhye, J. (2006). Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC). <http://www.ietf.org/rfc/rfc4342.txt>. Último acesso em Novembro 2011.
- [Huang et al. 1999] Huang, X., Sharma, R., and Keshav, S. (1999). The entrapid protocol development environment. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1107–1115 vol.3.
- [Jain 1991] Jain, R. (1991). *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley.
- [Jain et al. 1984] Jain, R. K., Chiu, D.-M. W., and Hawe, W. R. (1984). A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems. Technical report, Digital Equipment Corporation.
- [Jansen and Mcgregor 2005] Jansen, S. and Mcgregor, A. (2005). Simulation With Real World Network Stacks. In Kuhl, M. E., Steiger, N. M., Armstrong, F. B., and Joines, J. A., editors, *Winter Simulation Conference*, pages 2454–2463, Orlando, FL.
- [Jansen and McGregor 2006] Jansen, S. and McGregor, A. (2006). Performance, validation and testing with the network simulation cradle. In *Proceedings of the 14th IEEE International Symposium on Modeling, Analysis, and Simulation*, pages 355–362, Washington, DC, USA. IEEE Computer Society.
- [Jansen and McGregor 2008] Jansen, S. and McGregor, A. (2008). Network Simulation Cradle. http://www.nsnam.org/wiki/index.php/Network_Simulation_Cradle_Integration. Último acesso Novembro 2011.
- [Kohler et al. 2006] Kohler, E., Handley, M., and Floyd, S. (2006). Datagram Congestion Control Protocol (DCCP). <http://www.rfc-editor.org/rfc/rfc4340.txt>. Último acesso em Novembro 2011.
- [Kurose and Ross 2006] Kurose, J. F. and Ross, K. W. (2006). *Redes de Computadores e a Internet: Uma Abordagem Top-Down*. Addison Wesley, São Paulo, trad. 3 ed. edition.
- [Lai 2008] Lai, Y.-C. (2008). DCCP Congestion Control with Virtual Recovery to Achieve TCP-Fairness. *Communications Letters, IEEE*, 12(1):50–52.
- [NSNam 2011] NSNam (2011). Ns-3: Network simulation 3. <http://www.nsnam.org/>. Último acesso em Novembro 2011.
- [Perez and Ocampo 2009] Perez, J. C. and Ocampo, R. M. (2009). Implementation and evaluation of a dccp module for network simulator 2. In *Proc. TENCON 2009 - 2009 IEEE Region 10 Conf*, pages 1–6.
- [Zec 2003] Zec, M. (2003). Implementing a Clonable Network Stack in the FreeBSD Kernel. In *In Proceedings of the USENIX 2003 Annual Technical Conference*, pages 137–150.