

# Problemas e Soluções no Desenvolvimento de Componentes para o NS3: o Caso do DHCP

Andrey Blazejuk<sup>1</sup>, Alexsander Silva de Souza<sup>1</sup>, Sérgio Luis Cechin<sup>1</sup>

<sup>1</sup>Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)  
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{ablazejuk, asouza, cechin}@inf.ufrgs.br

**Abstract.** *Network simulations enable scientific research to create topologies that have similar behavior to real environments. Therefore, it allows researchers to develop new communication protocols, or to enhance protocols that are already consolidated in the area of computer networks. In this paper, strategies used by NS3 to model communication protocols and its applications in DHCP protocol implementation are shown. As a result, protocol implementation alternatives on NS3 will be discussed.*

**Resumo.** *A simulação de redes através de software possibilita a pesquisa científica com a reprodução de topologias que apresentem comportamentos semelhantes aos de um ambiente real. Dessa forma, permite que o pesquisador possa desenvolver novos protocolos de comunicação ou modelar protocolos já consolidados. Neste artigo são apresentadas as estratégias empregadas pelo NS3 para a modelagem de protocolos de comunicação e a aplicação das mesmas na implementação do protocolo DHCP. Como resultado serão discutidas as alternativas de implementação de protocolos no NS3.*

## 1. Introdução

O desenvolvimento de protocolos e sistemas para a área de redes requer especificações claras e implementações que possam ser verificadas em relação àquelas especificações. Essas implementações são complexas e de difícil depuração, considerando a sofisticação que as redes atingiram atualmente. Nesse cenário de alto custo de desenvolvimento e de verificação dessas implementações, tem-se utilizado com frequência a técnica de simulação. Essa técnica permite prever o comportamento de sistemas quando estes ainda estão indisponíveis, o que é comum durante o desenvolvimento. Mesmo quando o sistema está disponível, a simulação pode ser preferível por permitir a avaliação numa ampla gama de cargas de trabalho e ambientes [Jain 1991].

O projeto SDCN (*Software Defined Carrier Network*), uma parceria entre o Instituto de Informática da UFRGS e a Parks S/A Comunicações Digitais, utiliza o simulador NS3 (*Network Simulator 3*) [NS3 2015] para analisar o funcionamento de redes definidas em software. Apesar dos recursos oferecidos pela ferramenta, é comum não encontrar algum protocolo necessário. Isso acontece porque esses recursos são desenvolvidos segundo a demanda da comunidade de usuários, que podem não coincidir com os requisitos do projeto.

Nesse artigo serão apresentadas as dificuldades e discutidas as soluções adotadas para criar componentes no simulador NS3. Especificamente será discutido o desenvol-

vimento necessário para utilizar o protocolo DHCP (*Dynamic Host Configuration Protocol*) [Droms 1997]. Uma solução para a integração protocolo DHCP ao NS3 foi apresentada por Radu Lupu [Lupu 2011]. Essa implementação possui simplificações e limitações incompatíveis com os objetivos do projeto SDCN, o que motivou a criação de novos componentes. Entre os problemas existentes, destacam-se três limitações fundamentais para este artigo:

**Limitação 1** todas as mensagens DHCP mal formadas, que não atendem ao formato especificado pelo padrão por serem serializadas incorretamente, tornando o componente incompatível com outros modelos;

**Limitação 2** máquina de estados reduzida, gerando menor tráfego de mensagens;

**Limitação 3** artifícios empregados para contornar limitações do NS3, também gerando problemas de compatibilidade com outros componentes.

O artigo está organizado da seguinte forma. A Seção 2 apresenta o simulador, justificando sua escolha para o projeto SDCN. Na Seção 3, os aspectos relevantes do protocolo DHCP para a simulação são revistos. O modelo desenvolvido com os recursos fornecidos pelo simulador é apresentado na Seção 4, onde são enumeradas as dificuldades enfrentadas e as soluções empregadas. A Seção 5 mostra a validação dos componentes, verificando a correta operação do protocolo. Finalmente, a Seção 6 conclui o artigo, sugerindo a abordagem proposta como forma de desenvolvimento de novos componentes.

## 2. Apresentação do simulador

O NS3 é um simulador de redes usado principalmente para pesquisa e uso educacional. Encoraja a contribuição da comunidade científica ao possuir código aberto, documentação gerada a partir do código, facilidade de depuração e de análise e extração de resultados. A equipe de desenvolvimento do NS3 gera melhorias e correções de defeitos a cada três meses, diretamente no sítio da ferramenta.

A ferramenta é um simulador de eventos discretos, cujas descrições são feitas em linguagem C++ ou Python. Caso seja desejado, também possui um agendador de tarefas em tempo contínuo que permite a integração dos modelos simulados com equipamentos de rede reais.

O simulador define classes que representam nodos, dispositivos de rede, canais, aplicações e assistentes de topologia (chamados de *helpers*). Nodos podem ser vistos como computadores, aos quais funcionalidades são adicionadas. Dispositivos de rede são instalados em nodos para permitir a comunicação através dos canais. Canais de comunicação conectam dispositivos de rede, permitindo a definição de características tais como a taxa de transferência de dados e o atraso. Aplicações modelam funcionalidades que executam nos nodos. Os *helpers* permitem a simplificação de tarefas rotineiras durante a definição de descrições de simulação.

## 3. Funcionamento do protocolo DHCP

O protocolo DHCP é baseado no protocolo BOOTP (*Bootstrap Protocol*) [Croft and Gilmore 1985], acrescentando a capacidade de alocação automática de endereços reusáveis da rede e outras opções de configuração. Segue o modelo cliente-servidor, onde os servidores alocam endereços disponíveis da rede e configuram dinamicamente os clientes. O protocolo de transporte utilizado é o UDP. A porta 67 do

servidor é reservada para receber as mensagens do cliente e os pacotes enviados pelo servidor são recebidos na porta 68 do cliente.

A concessão de um endereço IP para um cliente DHCP ocorre da seguinte maneira:

1. O cliente envia uma mensagem do tipo DHCPDISCOVER em *broadcast* para encontrar servidores disponíveis.
2. Ao receber o pacote, o servidor reserva um endereço IP que ainda não esteja alocado na rede e oferece o mesmo ao cliente através de uma mensagem DHCPPOFFER, que pode ser em *broadcast* ou *unicast*, de acordo com o valor do campo *BOOTP flags* da mensagem recebida anteriormente.
3. A resposta do cliente vem com a mensagem DHCPREQUEST, onde o endereço oferecido em DHCPPOFFER é aceito e solicitado. O pacote é enviado em *broadcast*.
4. Para confirmar e concluir o processo, a mensagem DHCPACK é enviada ao cliente, informando o tempo de duração da concessão. Assim como DHCPPOFFER, este pacote pode ser enviado em *broadcast* ou *unicast*.

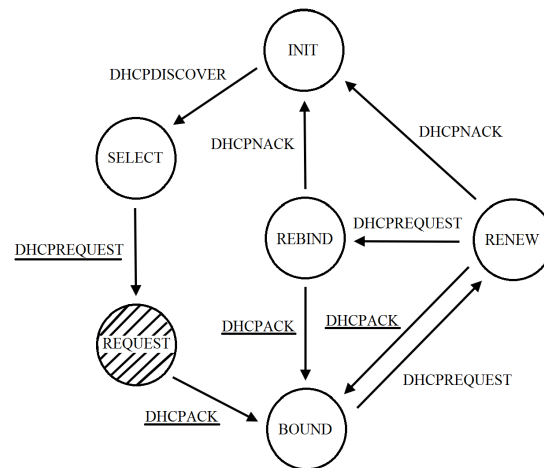
Para renovar seu endereço atual o cliente envia um pacote DHCPPOFFER em *unicast* para o servidor, que responde com um DHCPACK. A partir deste ponto ambas as partes consideram o tempo da concessão estendido.

#### 4. Modelagem do protocolo no simulador

A Limitação 1 apresentada na Introdução é causada pelas rotinas de serialização do cabeçalho DHCP, que não respeitavam o formato definido no padrão do IETF. O projeto SDCN requer que esse componente interaja com outros módulos, que esperam mensagens bem formadas. Esta característica também era desejada para permitir que os pacotes gerados sejam validados, utilizando softwares como o Wireshark [Foundation 2015].

A solução adotada por este artigo preenche todos os campos obrigatórios e acrescenta a opção *DHCP Message Type*, utilizada para determinar em qual etapa o processo de concessão se encontra. As duas mensagens enviadas pelo cliente contém a opção *Requested IP Address*, usada para que o cliente solicite determinado endereço IP ao servidor. Adicionalmente, DHCPREQUEST utiliza a opção *DHCP Server Identifier* para especificar de qual servidor DHCP o endereço oferecido é solicitado. Para ambas mensagens enviadas pelo servidor foram escolhidas as opções: *Subnet Mask* (máscara da rede parametrizada através do *helper* do servidor), *Renewal Time Value* (tempo até que o cliente envie a mensagem de DHCPPOFFER para renovar o empréstimo de seu endereço), *IP Address Lease Time* (tempo total, a partir da concessão, até que o IP se torne inválido) e *DHCP Server Identifier* (informando ao cliente qual o servidor que realizou o empréstimo). A opção *End* marca o fim da lista de opções DHCP em todas as mensagens.

O segundo problema, a máquina de estados muito simplificada (Limitação 2), provocava uma redução do número de mensagens trocadas durante a configuração de endereços IP. O processo de atribuição era feito apenas com o envio de uma mensagem DHCPDISCOVER e de sua resposta DHCPPOFFER, sem utilizar DHCPREQUEST e DHCPACK. A renovação do empréstimo do IP também era falha, pois não utilizava uma mensagem de confirmação DHCPACK. Uma das métricas avaliadas no projeto SDCN é o tráfego de protocolos de controle, onde essas simplificações tem um impacto direto.



**Figura 1. Máquina de estados do cliente DHCP. O estado hachurado e as mensagens sublinhadas foram adicionados pelo novo modelo.**

O componente desenvolvido para o projeto implementa uma máquina de estados mais próxima daquela prevista no protocolo (figura 1), utilizando as quatro mensagens. Alguns aspectos descritos na especificação do DHCP foram explicitamente ignorados, pois não interferem com os objetivos do projeto. Entre as abstrações feitas estão:

- O envio de pacotes do servidor para o cliente é sempre em *unicast*, mesmo quando o mesmo solicita o envio em *broadcast*.
- Apenas as opções fundamentais para a comunicação entre as partes são processadas em cada aplicação.
- Os tempos de concessão (4 segundos), renovação (metade do tempo de concessão) e retransmissão de pacotes (5 segundos) foram definidos como constantes para ambas aplicações.
- As mensagens DHCPNAK, DHCPDECLINE, DHCPRELEASE e DHCPINFORM não são utilizadas.

Finalmente, a solução de Lupu necessita de artifícios não usuais para contornar uma premissa do NS3, como descrito pela Limitação 3. O simulador assume que toda interface de rede compatível com o protocolo IP terá um endereço definido antes da simulação iniciar. Essa limitação é incompatível com o objetivo do projeto de avaliar o funcionamento de protocolos de gerência de rede, que podem alterar a configuração da rede dinamicamente. Lupu requer que atribua-se endereços provisórios a todas as interfaces, o que polui a tabela de roteamento do simulador com prefixos de rede inexistentes. A nova solução tirou proveito da natureza de software livre do NS3, e modificou o próprio simulador para remover essa premissa, evitando o uso de artifícios não intuitivos na descrição da topologia.

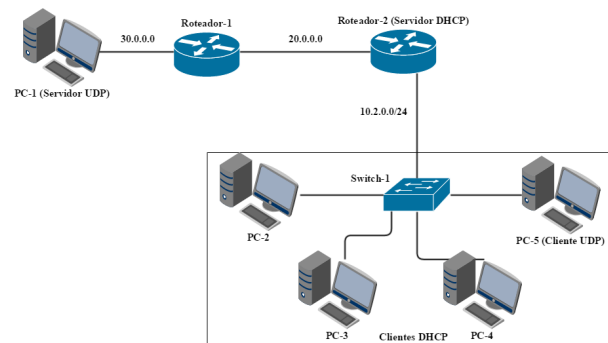


Figura 2. Topologia de teste.

## 5. Resultados obtidos

Para validar o módulo desenvolvido foi utilizada a topologia apresentada na figura 2. Nessa topologia é possível observar como esse módulo pode ser combinado com outros componentes disponíveis no NS3. Por exemplo, um método para gerar tabelas de roteamento atualizadas durante a simulação, uma aplicação que envia pacotes UDP e outra que os recebe (executadas no PC-5 e no PC-1, respectivamente).

O simulador permite a captura do tráfego de pacotes dos dispositivos de rede selecionados, gerando arquivos com o formato PCAP. As limitações do modelo de Lupu impediam uma validação externa do traço de execução devido ao formato incorreto das mensagens. O modelo criado com cabeçalhos adaptados para não só permitir a troca de informações entre as aplicações, mas também atender o formato previsto pelo IETF, gera um traço de execução que foi reconhecido com sucesso pelo software Wireshark, como mostrado na figura 3.

Na nova implementação a máquina de estados do protocolo utiliza as quatro principais mensagens do protocolo para configuração do cliente. Em destaque na figura 3, é possível observar as mensagens que estavam ausentes na implementação de Lupu, e que impactavam na análise do tráfego de gerenciamento no projeto SDCN.

Finalmente, antes das alterações realizadas no NS3 exigia-se que roteadores sempre tivessem endereço IP definido. O descumprimento dessa premissa gerava um colapso do software durante a execução de seu processo de resolução de rotas. Com a nova implementação é possível que uma interface com capacidade IP não possua nenhum endereço configurado, permitindo a modelagem de uma gama de protocolos de configuração dinâmica.

## 6. Conclusão

O modelo criado fornece um módulo para o simulador NS3 que pode ser usado em topologias que utilizem o protocolo DHCP. Suas características, em relação ao que existia anteriormente, permitem representar melhor o funcionamento de dispositivos reais que suportam o protocolo para que o comportamento das simulações realizadas seja mais semelhante ao de uma rede física.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	0.0.0.0	255.255.255.255	DHCP	DHCP Discover - Transaction ID 0x67458b6b
2	0.000062	10.2.0.1	10.2.0.254	DHCP	DHCP Offer - Transaction ID 0x67458b6b
3	0.000062	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0xc6237b32
4	0.000127	10.2.0.1	10.2.0.254	DHCP	DHCP ACK - Transaction ID 0xc6237b32
7	1.004024	10.2.0.254	30.0.0.2	UDP	Source port: 49153 Destination port: 10
10	1.018980	30.0.0.2	10.2.0.254	UDP	Source port: 10 Destination port: 49153
11	1.500000	10.2.0.254	30.0.0.2	UDP	Source port: 49153 Destination port: 10
12	1.514929	30.0.0.2	10.2.0.254	UDP	Source port: 10 Destination port: 49153
13	2.000000	10.2.0.254	30.0.0.2	UDP	Source port: 49153 Destination port: 10
14	2.000127	10.2.0.254	10.2.0.1	DHCP	DHCP Request - Transaction ID 0x69983c64
15	2.000190	10.2.0.1	10.2.0.254	DHCP	DHCP ACK - Transaction ID 0x69983c64
16	2.014929	30.0.0.2	10.2.0.254	UDP	Source port: 10 Destination port: 49153
17	2.500000	10.2.0.254	30.0.0.2	UDP	Source port: 49153 Destination port: 10
18	2.514929	30.0.0.2	10.2.0.254	UDP	Source port: 10 Destination port: 49153
19	3.000000	10.2.0.254	30.0.0.2	UDP	Source port: 49153 Destination port: 10
20	3.014929	30.0.0.2	10.2.0.254	UDP	Source port: 10 Destination port: 49153

Figura 3. Traço de execução da simulação.

O trabalho apresentado serve como referência para a modelagem de outros protocolos, que podem ser implementados em qualquer simulador de redes, de acordo com a preferência ou necessidade do desenvolvedor. A partir do módulo criado e do resultado final, pode-se deduzir que o processo descrito foi efetivo e tem sua utilidade para a pesquisa, ilustrando o comportamento de redes reais através de simulações.

## Referências

- Croft, W. and Gilmore, J. (1985). Bootstrap protocol. RFC 951, RFC Editor. <http://www.rfc-editor.org/rfc/rfc951.txt>.
- Droms, R. (1997). Dynamic host configuration protocol. RFC 2131, RFC Editor. <http://www.rfc-editor.org/rfc/rfc2131.txt>.
- Foundation, W. (2015). What is wireshark? <https://www.wireshark.org/faq.html#q1.1>. [Online; acesso em 10-Julho-2015].
- Jain, R. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley-Interscience, 1st edition.
- Lupu, R. C. (2011). Dhcp client-server. <http://www.elcom.pub.ro/~rlupu/>. [Online; acesso em 10-Julho-2015].
- NS3 (2015). What is ns3. <http://www.nsnam.org/overview/what-is-ns-3/>. [Online; acesso em 10-Julho-2015].