



**Universidade:  
presente!**

**UFRGS**  
PROPEAQ



**XXXI SIC**

21. 25. OUTUBRO. CAMPUS DO VALE

<b>Evento</b>	Salão UFRGS 2019: SIC - XXXI SALÃO DE INICIAÇÃO CIENTÍFICA DA UFRGS
<b>Ano</b>	2019
<b>Local</b>	Campus do Vale - UFRGS
<b>Título</b>	Automatizando a geração de device drivers para GNU/Linux a partir de especificações P4
<b>Autor</b>	YURI AUGUSTO DUTRA JASCHEK
<b>Orientador</b>	WEVERTON LUIS DA COSTA CORDEIRO

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA

AUTOMATIZANDO A GERAÇÃO DE DEVICE DRIVERS PARA GNU/LINUX A  
PARTIR DE ESPECIFICAÇÕES P4

Autor: YURI JASCHEK

Orientador: WEVERTON CORDEIRO

### Introdução

O objetivo da bolsa de pesquisa é desenvolver algoritmos para geração automatizada de *device drivers* GNU/Linux a partir de especificações de protocolos de redes de computadores escritos em P4 [1]. Como P4 é uma linguagem de domínio específico, as estruturas de dados a serem utilizadas devem ser propícias para protocolos de rede e também serem disponibilizadas pelo *kernel* Linux, uma vez que estes são executados no modo *kernel*.

### Metodologia

A linguagem P4 é usada para especificar o formato de pacotes de dados e a forma como os mesmos serão processados pelos dispositivos de encaminhamento da rede. Para esse fim, é necessário que não apenas o núcleo da rede “fale” o protocolo implementado em P4, mas também as bordas, ou seja, os servidores e estações de usuários (entre outros). Para isso, o *kernel* das estações que ficam na borda da rede precisam implementar a pilha de protocolos “falada” na rede. Mais ainda, essa implementação deve permitir a troca de dados de forma eficiente *user space* e *kernel space*.

No escopo do projeto, está sendo desenvolvido um módulo para extrair as estruturas de pacotes de protocolos escritos em P4, e gerar *device drivers* que implementem esses protocolos. Para permitir que aplicações do *user space* comuniquem dados usando esses protocolos, foi escolhida a família de *sockets netlink*, devido a sua característica assíncrona de comunicação e também por já ser bastante utilizada em protocolos específicos de rede [2].

O processo de tradução de uma especificação P4 para uma estrutura com campos adequados ao protocolo se dará com um passo intermediário, de P4 para um código na linguagem C [3] e com a subsequente incorporação ao código responsável pela gerência do *device driver*. O foco atual é gerar *drivers* para o *kernel* Linux.

### Resultados preliminares

A automatização do processo torna praticável que qualquer aplicação GNU/Linux comunique dados usando quaisquer protocolos da pilha em uso. Considerando a execução no *kernel space*, espera-se obter desempenho satisfatório e similar a algumas implementações de *drivers* da pilha TCP/IP.

### Referências

- [1] Bosshart, Pat, et al. "P4: Programming protocol-independent packet processors." *ACM SIGCOMM Computer Communication Review* 44.3 (2014): 87-95.
- [2] Salim, J., et al. *Linux netlink as an ip services protocol*. No. RFC 3549. 2003.
- [3] L. Freire, M. Neves, L. Leal, K. Levchenko, A. Schaeffer-Filho, M. Barcellos. Uncovering Bugs in P4 Programs with Assertion-based Verification. In *Symposium on SDN Research (SOSR '18)*. ACM, New York, NY, USA, Article 4, 7 pages.