



دانشگاه تهران  
پردیس دانشکده‌های فنی  
دانشکده مهندسی برق و کامپیوتر



# توضیح خطا در شبکه‌های مبتنی بر نرم‌افزار با استفاده از استدلال مبتنی بر علیت

پایان‌نامه برای دریافت درجه کارشناسی ارشد در رشته مهندسی کامپیوتر  
گرایش نرم‌افزار

امیرحسین صیحانی

اساتید راهنما

دکتر حسین حجت و دکتر محمدرضا موسوی

شهریور ۱۴۰۱



چکیده

واژگان کلیدی



# فهرست مطالب

فصل ۱: دانش پیش زمینه	۳
۱.۱ مقدمه	۳
۲.۱ نکت پویا	۳
۱.۲.۱ دستور زبان نکت پویا	۴
۲.۲.۱ معنای عملیاتی نکت پویا	۴
۳.۲.۱ توصیف برنامه‌ها در نکت پویا	۶
مراجع	اول
واژه‌نامه فارسی به انگلیسی	سوم
واژه‌نامه انگلیسی به فارسی	پنجم

## فهرست کارهای باقیمانده



# فصل ۱

## دانش پیش زمینه

### ۱.۱ مقدمه

در این فصل مفاهیم مورد نیاز و استفاده در این پروژه مورد بررسی قرار می گیرند. این فصل، محل شرح کامل روش تحقیق است و بسته به نوع روش تحقیق و با نظر استاد راهنما می تواند «مواد و روش ها»<sup>۱</sup> نیز نام بگیرد. این فصل حدود ۱۵ صفحه است.

### ۲.۱ نتکت پویا

نتکت پویا<sup>۲</sup> برای رفع برخی از کاستی های نتکت ارائه شده است [۱]. به صورت دقیق تر نتکت پویا، امکان توصیف به روزرسانی سیاست های شبکه و همچنین رفتار شبکه در مقابل چندین بسته را ممکن می سازد.

---

<sup>۱</sup>Materials and Methods

<sup>۲</sup>DyNetKAT



## ۱.۲.۱ دستور زبان نت‌کت پویا

در نت‌کت پویا، از رفتار انتها به انتهای توصیف‌های شبکه در قالب عبارت‌های نت‌کت استفاده می‌شود. به همین منظور سینتکس نت‌کت پویا به صورت زیر تعریف می‌شود:

$$N ::= \text{NetKAT}^{-dup}$$

$$D ::= \perp | N; D | x?N; D | x!N; D | D \parallel DD \oplus D | X$$

$$X \triangleq D$$

در سینتکس بالا  $\text{NetKAT}^{-dup}$  قسمتی از زبان نت‌کت است که عبارت‌های  $dup$  از آن حذف شده است. عبارت‌های  $dup$  در توصیف‌های نت‌کت تاثیری در رفتار یک عبارت ندارند و هدف از استفاده از آن‌ها ثبت یک اثر از هر بسته پس از پردازش توسط یکی از عناصر شبکه است و امکان استدلال بر روی رفتار شبکه را ممکن می‌سازد. با توجه به این که در نت‌کت پویا رفتار انتها به انتهای یک عبارت نت‌کت مورد استفاده است، عبارت  $dup$  از دستور زبان کنار گذاشته شده است. نت‌کت پویا یک لیست از بسته‌های ورودی را پردازش می‌کند و یک لیست از مجموعه‌ی بسته‌های خروجی تولید می‌کند. اپراتور ترکیب متوالی  $N; D$ <sup>۳</sup> باعث می‌شود که یک بسته از لیست بسته‌های ورودی توسط سیاست  $N$  پردازش شود و سپس بسته‌ی توسط عبارت  $D$  پردازش می‌شود. در نت‌کت پویا امکان ارتباط توسط عبارت‌هایی به شکل  $x!N$  و  $x?N$  توصیف می‌شوند که به ترتیب ارسال و دریافت یک عبارت نت‌کت را روی کانال  $x$  توصیف می‌کنند. ترکیب موازی<sup>۴</sup> دو عبارت توسط  $D \parallel D$  توصیف می‌شود. در نهایت رفتارهای غیرقطعی<sup>۵</sup> توسط عبارت‌هایی به شکل  $D \oplus D$  توصیف می‌شوند.

## ۲.۲.۱ معنای عملیاتی نت‌کت پویا

معنای عملیاتی<sup>۶</sup> نت‌کت پویا با استفاده از عبارت‌هایی به شکل  $(d, H, H')$  تعریف می‌شوند که  $d$  عبارت نت‌کت پویا فعلی است،  $H$  لیست بسته‌هایی که در ادامه باید پردازش شوند و  $H'$  لیست بسته‌هایی است که به

<sup>۳</sup>Sequential Composition

<sup>۴</sup>Parallel Composition

<sup>۵</sup>Non-Deterministic

<sup>۶</sup>Operational Semantics

صورت موفقیت‌آمیز توسط شبکه پردازش شده‌اند. در اینجا فرض می‌شود که  $F = \{f_1, \dots, f_n\}$  یک مجموعه از فیلدهای بسته‌ها است. یک بسته به شکل یک تابع  $F \rightarrow \mathbb{N}$  توصیف می‌شود. برای یک بسته مانند  $\sigma$  تساوی  $\sigma(f_i) = v_i$  بیان می‌کند که مقدار فیلد  $f_i$  در بسته  $\sigma$  برابر با  $v_i$  است. یک لیست خالی از بسته‌ها با  $()$  نمایش داده می‌شود. اگر  $l$  یک لیست از بسته‌ها باشد  $l :: e$  لیستی است که حاصل از اضافه کردن بسته  $\sigma$  به ابتدای لیست به دست می‌آید. برچسب هر قانون که با  $\gamma$  مشخص می‌شود به صورت یکی از شکل‌های  $x!q, x?q, (\sigma, \sigma')$  یا  $rcfg(x, q)$  تعریف می‌شود که  $rcfg(x, q)$  به معنی انجام شدن  $x!q$  و  $x?q$  به صورت همگام<sup>۷</sup> است. قوانین زیر معنای عملیاتی نت‌کت پویا را تعریف می‌کنند:

$$(cpol_{\prec}) \frac{\sigma' \in \llbracket p \rrbracket (\sigma :: \langle \rangle)}{(p; q, \sigma :: H, H') \xrightarrow{(\sigma, \sigma')} (q, H, \sigma' :: H')} \quad (1.1)$$

$$(cpol_X) \frac{(p, H_0, H_1) \xrightarrow{\gamma} (p', H'_0, H'_1)}{(X, H_0, H_1) \xrightarrow{\gamma} (p', H'_0, H'_1)} X \triangleq p \quad (2.1)$$

$$(cpol_{\oplus}) \frac{(p, H_0, H'_0) \xrightarrow{\gamma} (p', H_1, H'_1)}{(p \oplus q, H_0, H'_0) \xrightarrow{\gamma} (p', H_1, H'_1)} \quad (3.1)$$

$$(cpol_{\oplus}) \frac{(q, H_0, H'_0) \xrightarrow{\gamma} (q', H_1, H'_1)}{(p \oplus q, H_0, H'_0) \xrightarrow{\gamma} (p', H_1, H'_1)} \quad (4.1)$$

$$(cpol_{\parallel}) \frac{(p, H_0, H'_0) \xrightarrow{\gamma} (p', H_1, H'_1)}{(p \parallel q, H_0, H'_0) \xrightarrow{\gamma} (p' \parallel q, H_1, H'_1)} \quad (5.1)$$

$$(cpol_{\parallel}) \frac{(q, H_0, H'_0) \xrightarrow{\gamma} (q', H_1, H'_1)}{(p \parallel q, H_0, H'_0) \xrightarrow{\gamma} (p \parallel q', H_1, H'_1)} \quad (6.1)$$

$$(cpol_{?}) \frac{}{(x?p; q, H, H') \xrightarrow{x?p} (q, H, H')} \quad (7.1)$$

$$(cpol_{!}) \frac{}{(x!p; q, H, H') \xrightarrow{x!p} (q, H, H')} \quad (8.1)$$

$$(cpol_{!?}) \frac{(q, H, H') \xrightarrow{x!p} (q', H, H') (s, H, H') \xrightarrow{x?p} (s', H, H')}{(q \parallel, H, H') \xrightarrow{rcfg(x,p)} (q' \parallel s', H, H')} \quad (9.1)$$

$$(cpol_{?!}) \frac{(q, H, H') \xrightarrow{x?p} (q', H, H') (s, H, H') \xrightarrow{x!p} (s', H, H')}{(q \parallel, H, H') \xrightarrow{rcfg(x,p)} (q' \parallel s', H, H')} \quad (10.1)$$

<sup>7</sup>Synchronized



شکل ۱.۱: مثال دیوار آتش

قانون ۱.۱ انجام یک عملیات مانند  $(\sigma, \sigma')$  که به معنای پردازش بسته‌ی ابتدایی لیست ورودی توسط عبارت  $p$  و افزودن خروجی حاصل از آن مانند  $\sigma'$  به لیست خروجی است را مشخص می‌کند. قانون ۲.۱ بیان می‌کند که رفتار متغیر  $X$  که برابر با عبارت  $p$  است معادل با رفتار عبارت  $p$  است. قوانین ۳.۱ و ۴.۱ رفتار غیرقطعی را توصیف می‌کنند. قوانین ۵.۱ و ۶.۱ رفتار دو عبارت موازی را توصیف می‌کنند. قوانین ۷.۱ و ۸.۱ مشخص می‌کنند که ارسال یا دریافت پیام در نکت پویا پردازشی روی بسته‌ها انجام نمی‌دهد. در نهایت همگام‌سازی<sup>۸</sup> ارسال و دریافت پیام توسط قوانین ۷.۱ و ۸.۱ توصیف شده است.

### ۳.۲.۱ توصیف برنامه‌ها در نکت پویا

در ادامه چگونگی توصیف یک دیوار آتش<sup>۹</sup> وابسته به حالت<sup>۱۰</sup> با استفاده از نکت پویا بیان می‌شود. شبکه‌ی شکل ۱.۱ را در نظر بگیرید. در این شبکه هدف این است که امکان ارتباط از داخل شبکه به بیرون فراهم باشد ولی امکان ارسال بسته از خارج شبکه ممکن نباشد. اما زمانی که یک بسته به خارج شبکه ارسال شد، دیوار آتش باید اجازه‌ی عبور بسته‌ها از بیرون را بدهد تا پاسخ بسته‌ها دریافت شوند. برای توصیف این شبکه می‌توان از عبارت

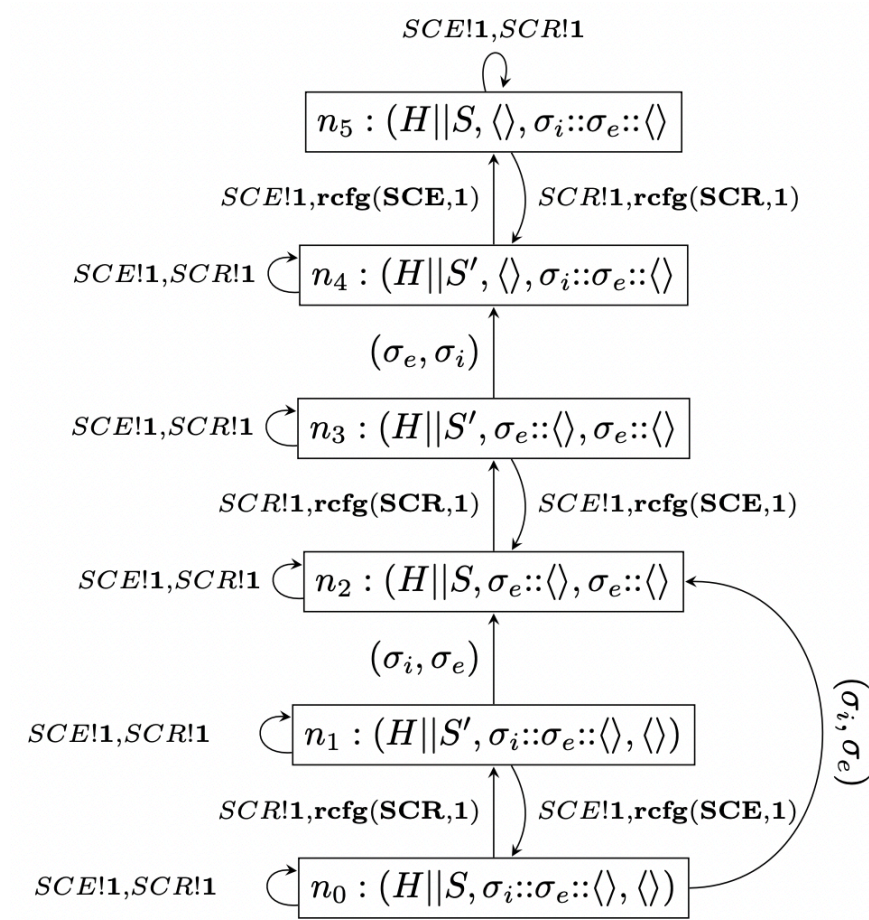
<sup>۸</sup>Synchronization<sup>۹</sup>Firewall<sup>۱۰</sup>Stateful

نت‌کت پویای زیر استفاده کرد:

$$\begin{aligned}
 Host &\triangleq secConReq!1; Host \oplus secConEnd!1; Host \\
 Switch &\triangleq (port = int) \cdot (port \leftarrow ext); Switch \oplus \\
 &\quad (port = ext) \cdot 0; Switch \oplus \\
 &\quad secConReq?1; Switch' \\
 Switch' &\triangleq (port = int) \cdot (port \leftarrow ext); Switch' \oplus \\
 &\quad (port = ext) \cdot (port \leftarrow int); Switch' \oplus \\
 &\quad secConEnd?1; Switch \\
 Init &\triangleq Host \parallel Switch
 \end{aligned}$$

در این توصیف عملیات  $secConReq$  برای شروع ارتباط امن و  $secConEnd$  برای خاتمه‌ی ارتباط امن در نظر گرفته شده‌است. بنابراین برنامه‌ی سوییچ پس از دریافت پیام برای شروع ارتباط امن تبدیل به برنامه‌ی  $Switch'$  می‌شود که در آن اجازه‌ی ارسال بسته از پورت خارجی به پورت داخلی را دارد. پس از دریافت پیام برای خاتمه‌ی ارتباط امن، برنامه به حالت اولیه‌ی خود بر می‌گردد و دوباره تمامی بسته‌های ورودی از پورت خارجی را رها می‌کند. در نهایت رفتار کل شبکه با استفاده از ترکیب موازی یک هاست و یک سوییچ در حالت اولیه خود توصیف می‌شود. نمودار نمایش داده شده در شکل ۲.۱ سیستم انتقال برچسب‌دار<sup>۱۱</sup> این شبکه را در حالتی که یک بسته روی پورت ورودی و یک بسته روی پورت خروجی شبکه وجود دارد نشان می‌دهد. همانطور که در نمودار مشخص است، عملیات  $(\sigma_e, \sigma_i)$  که به معنای ارسال بسته از پورت ورودی به پورت خروجی است تنها در قسمتی از این سیستم انتقال قابل دسترسی است که پیش از آن یکی از عملیات‌های  $SCR?1$  یا  $rcfg(SCR, 1)$  انجام شده باشند. بنابراین در این حالت شبکه تنها در صورتی که بسته خارجی را به داخل ارسال می‌کند که پیش از آن پیام آغاز ارتباط امن دریافت کرده باشد.

<sup>11</sup>Labeled Transition System



شکل ۲.۱: سیستم انتقال برچسب‌دار برای شبکه‌ی دیوار آتش

## مراجع

- [1] Caltais, Georgiana, Hojjat, Hossein, Mousavi, Mohammad, and Tunc, Hunkar Can. Dynetkat: An algebra of dynamic networks. *arXiv preprint arXiv:2102.10035*, 2021. [3](#)



# واژه‌نامهٔ فارسی به انگلیسی





# واژه‌نامه انگلیسی به فارسی



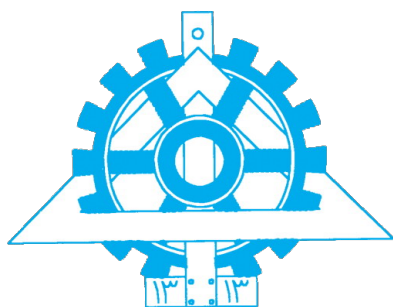


## Abstract

This thesis studies on writing projects, theses and dissertations using tehran-thesis class. It ...

**Keywords** Writing Thesis, Template, L<sup>A</sup>T<sub>E</sub>X, X<sub>Y</sub>Y Persian





University of Tehran  
College of Engineering  
Faculty of Electrical and  
Computer Engineering



# Explaining Failures in Software-Defined Networks Using Casual Reasoning

A Thesis submitted to the Graduate Studies Office  
In partial fulfillment of the requirements for  
The degree of Master of Science  
in Computer Engineering - Software Engineering

By:

**Amir Hossein Seyhani**

Supervisors:

**Dr. Hossein Hojjat and Dr. Mohammad Reza Mousavi**

Advisor:

**First Advisor**

September 2022