1 Project Description

This project will create an Integrated Development Environment (IDE) for the network programming language, P4 [1]. Specifically, I will develop a plugin for Microsoft VSCode that includes: (i) a syntactic analyzer, (ii) a semantic analyzer for basic errors and usage (e.g., type checker and resource estimator), and (iii) functionality to protect proprietary information.

1.1 Background

Over the last several years, networks have become increasingly programmable. This innovation has the potential to completely change the way companies and customers purchase, customize, and use network technology.

The ability to program switches has numerous benefits. It allows users to tailor the network device to their particular needs. It allows changes to be made at the speed of software development, rather than hardware development. And, it allows for upgrades and bug fixes completely in software.

While the concept of network programming is extremely advantageous in theory, since it is a relatively new concept, there are very few tools available that deal with the development process. Additionally, networking programming languages, such as P4 [1], are more restricted than other languages, which make it more difficult to work with.

For example, a vendor like Cisco might develop a program to run their switch titled switch.p4, and needs to share this program with their customers without revealing their implementation. Ideally, customers would be able to change a program through their own separate file, user.p4, without altering anything in Cisco’s program (to avoid "breaking" Cisco’s code). However, unlike C, P4 does not allow for binary distributions, nor does P4 have a module system. Because of this, we propose to develop an IDE that “hides” proprietary information while allowing for customization.

Project Goal. This project will create a tool which network operators can use to assist with programming these switches while still preserving some of the proprietary features
that the switches hold from the manufacturer and ensuring that these features cannot be compromised in any way.

One way to achieve this goal is to have a certain set of files hosted on the company’s end, including a generic framework (see switch.p4 in the diagram below) and a compiler.

The client receives this information and makes it easier for the programmer to add whatever code necessary. The client then sends this combined information back to the server, where it combines the two parts and compiles the program.

The server side will combine user.p4 and switch.p4 by first exposing “hooks” where users can add their extensions to the switch.p4 program. The client will then communicate with the server in order to learn where exactly these “hooks” are, and reveal them to the user so that he can then edit the user.p4 program properly.

The diagram below offers a visual representation of the process:

![Diagram](image)

## 2 Prior Solutions

This design is inspired by prior work on Aspect-Oriented programming [2].

## 3 Current Status

Over the summer, I worked with Prof. Soulé on some initial prototyping techniques while learning some basic concepts from him through weekly meetings. So far, I have written a grammar for the P4 language, and used the ANTLR tool to generate a parser. I have also begun to implement basic semantic analyzers using the visitor pattern to traverse the parse tree.

## 4 Planned Progress

**Personal Area of Focus.** I will be working on a client side plugin for Microsoft VSCode. My responsibilities include:

- Locating and identifying syntax errors.
• Locating and identifying semantic errors. (e.g., type checking)
• Estimating resource consumption.
• Providing autocompletion.
• Communicating with the server component, which will be developed by Ali.

My work will also expose me to a variety of different languages, environments, and development tools that will not only supplement programming techniques learned in prior classes (such as CPSC 223), but introduce me to situations and experiences that I would not be able to learn in a classroom setting. These experiences include:

• Use of Git and Github.
• Use of tools such as ANTLR and Latex.
• Weekly conference calls using Cisco Webex with team members.

5 Collaborators

For this project, I will be supervised by Prof. Robert Soulé. I will also collaborate with Dr. Mario Baldi from Cisco, Prof. Antonio Carzaniga from USI, and Ali Fattaholmanan, a PhD student of Prof. Soulé.

References
