I/UCRC Executive Summary – Project Synopsis

**Project Title:** Leveraging Rigorous Software Specification and Testing to Developing and Orchestrating Reliable Applications for Network Softwarization

| **Presentation Date:** June 19th, 2019 | **Presentation Type:** New (New, Status, Final) | **Project Budget:** $35,000 |
| **Center:** Security & Software Engineering Research Center | **Center Director:** Wayne Zage |
| **Site (University):** Ball State University | **Co-Director:** |
| **Project PI:** Xin Sun, Lan Lin | **Email:** xsun6@bsu.edu, llin4@bsu.edu | **Phone:** 786-200-0099 |

**Project description:**
The trend of softwarizing cyber infrastructures has posed new challenges to software engineering. While traditional software effects a single device, emerging software (in a softwarized cyber infrastructure) typically manages a large number of hardware boxes (such as networking and security devices) that collectively determine the network behavior. Correct functioning of such software relies on precise analysis of the current network state, which has not been adequately studied and understood.

This proposal strives to fill in this gap by exploring a holistic modeling framework for systematically developing and analyzing software applications for the Software-Defined Networking (SDN) paradigm. On the one hand, the applications dictate the network behavior; on the other hand, they may not have comprehensive knowledge of the network state, and may interact with each other in unpredictable ways, both leading to incorrect network behavior.

We propose to leverage rigorous software specification and testing methods to a theoretical framework that serves two purposes: (1) to formally model not only the software states but also the effect of software on the network states, in the specification, design, and testing of SDN apps, and (2) to formally model the interactions of multiple SDN apps running in parallel on the same network for precise analysis.

We observe the potential to extend this formal treatment to include other software verification and validation techniques, to ensure high-confidence software-defined networks in the field with provable end-to-end network service assurance.

**Experimental plan:**
Our planned investigation of the proposed framework has two phases:

- **Phase I:** We will explore how rigorous specification, design, and testing methodologies could be utilized in a systematic process to develop new SDN apps or migrate existing algorithms and protocols to the SDN environment. Our preliminary work in the migration of the MAC learning switch algorithm has identified the potential benefit of this approach over purely ad-hoc heuristic development in two aspects: (1) it exposes the change of the system boundary as a result of the SDN architectural changes, and (2) it precisely defines the effect of software on the network states.

- **Phase II:** We will explore how rigorous specification could be utilized to model and analyze the interactions of multiple SDN apps, and to detect any potential conflicts and identify the root causes. Our preliminary work has identified a potential conflict as a result of running a load-balancing app and a power-saving app in parallel, which may route flows back and forth between alternative paths and destabilize the network.

We will complete the theory development for our proposed framework, and demonstrate its feasibility and applicability with multiple simplified but essential and widely-deployed SDN apps.

We will also discuss our approach with the S2ERC affiliates; if any affiliate is currently running multiple SDN apps, we will work with them to try to apply our approach to their deployment, as a way to validate our approach.

**Related work elsewhere and/or in the center:**
Much of the prior effort in SDN focused on developing algorithms for individual SDN apps. To the best of our knowledge little work has been done focusing on applying software engineering methodologies to the production of SDN apps. More recently the problem of control conflicts caused by running multiple independently-developed SDN apps, and the resulting destabilization of the network, has started to receive attention from the networking community.

Corybantic (J. Mogul et al, 2013) and Athens (A. AuYoung, et al, 2014) take a coarse-grained approach that resolves potential conflicts at run-time (i.e., when the network is in operation). They let individual SDN apps generate “proposals” for network configuration changes, and then require each app to evaluate all proposals, based on some predefined policy or voting mechanism. While this approach can successfully resolve conflicts, it does so...
with significant costs: (i) the approach requires all SDN apps to implement the additional functionality of generating and evaluating proposals; (ii) it only selects a single proposal at a time, which can leave out potentially better solutions that combine multiple conflict-free proposals, leading to sub-optimal network configurations; (iii) this approach does not expose the root cause of the conflicts and thus it can only resolve, but not avoid, the conflicts, which again raises concerns on efficiency and optimality.

Another prior work (A. Bairley and G. Xie, 2016) takes a similar approach (and thus suffers from similar drawbacks), except that it seeks to combine multiple proposals to form a globally-optimal configuration - it does so by modeling the problem as a multi-objective optimization one and using an evolutionary approach to solve it.

Other prior work such as VeriFlow (A Khurshid, et al, 2013) and NetPlumber (P. Kazemian, et. al, 2013) concern individual-flow-level forwarding behavior and can detect policy violations (such as black holes and loops) at that level; however, none of the work can detect the network state oscillation caused by conflicts of running multiple SDN apps, since state oscillation is not a violation of flow policy (it is a higher-level issue).

### How this project differs from related work:

Our proposed approach is different from previous work in leveraging software engineering methodologies to (1) systematically developing SDN apps with high quality, and (2) formally and precisely modeling individual SDN apps and their interactions. We will produce SDN apps following rigorous software specification, design, and testing, and develop models that will be capable of answering "what-if" questions about running multiple SDN apps before they are deployed. This will help network managers select compatible SDN apps and avoid any potential conflict at runtime. This approach also does not require any modification to existing SDN apps, making it practical to deploy.

In contrast, the prior work has none of these properties; and we believe these properties are the key to a more relevant, more useful, and more deployable solution.

### Milestones planned for the upcoming year:

- **Milestone I:** SDN app development using software engineering methodologies
- **Milestone II:** development of formal models for individual SDN apps
- **Milestone III:** development of a formal model that captures the interaction among multiple SDN apps
- **Milestone IV:** creation of algorithmic approaches for computing the model and for automatic detection of conflicts based on the model

### Deliverables planned for the upcoming year:

- formal models for individual SDN apps and for capturing the interaction of multiple SDN apps
- an algorithmic approach for computing the models
- an algorithmic approach for automatic detection of conflicts based on the model
- requirements, design, testing documentations and implementations of a number of SDN apps used in the case studies

### How the project may be transformative and/or benefit society:

- This project transforms the way we build and run software-defined networks, by introducing a theoretical tool for developing SDN apps as well as detecting and resolving control conflicts.
- This project fosters innovation by solving a key problem in the emerging SDN paradigm.
- This project makes our cyber infrastructures more resilient and more secure.

### Research areas of expertise needed for project success:


### Potential member company benefits:

The theory, models, and algorithms developed in this project can directly benefit any company that has an SDN-based infrastructure or is interested in deploying SDN in the future.

### Progress to date:

We have identified a case where a load-balancing app and a power-saving app will cause conflict if running together; the conflict will destabilize the network, as it causes the network to oscillate between two states. We have also looked at how to migrate the MAC learning switch algorithm to the SDN environment using software engineering methodologies.

### Project Start Date: 11/1/2019  |  Estimated Knowledge Transfer Date: 10/31/2020

The Executive Summary is used by corporate stakeholders in evaluating the value of their leveraged investment in the center and its projects. It also enables stakeholders to discuss and decide on the projects that provide value to their respective organizations. Ideally, the tool is completed and shared in advance of IAB meetings to help enable rational decision making.