**Ravel: Orchestrating Software-Defined Networks**

**Anduo Wang  Brighten Godfrey  Matthew Caesar (University of Illinois Urbana-Champaign)**

**Ravel design**
- SDN network is collectively driven by many applications
- the applications interact
  - applications collaborate, e.g., access control and routing
  - applications conflict, e.g., access control and load balancer
- existing solution unsatisfying
  - require a master program that coordinates the dynamics

**Ravel components**
- users: control program embedded with SQL query and update
- views abstraction: created, queried, and updated by the applications
  - programmable: derived SQL view
  - open: SQL view is readily available to others without re-compilation
- base tables: store network state, hides hardware detail, fast network access and update

**Ravel services**
- vertical orchestration
  - synchronize derived views and their source views / tables
  - enables network control via view update
- horizontal orchestration
  - a priority-based data-sharing protocol that coordinates view updates
  - allows applications act autonomously while living in harmony

**Motivation**
- upon new tenant flow request, install a load-balanced, safe route

**Scenario 1: Upon link failure, re route**
- check broken path, re-route
- orchestrated updates: re route via (172, 38)

**Scenario 2: Upon new tenant flow request, install a load-balanced, safe route**
- install a load-balanced, safe route
- orchestrated updates: install alternative route that is load-balanced and safe