Computer Science Department Graduate Programs Infosession & Orientation

Dr. Nahapetian (Dr. N)

csgrad@csun.edu

http://www.ecs.csun.edu/csgrad

Plan for Today

- Program Requirements
- Thesis Work
- Funding Opportunities
- Admissions Information
- Program Prerequisites
- Q/A Session

Computer Science MS

Breadth Requirement– Choose 4

- COMP 529/L Advanced Network Topics and Lab
- COMP 610 Data Structures and Algorithms
- COMP 615 Advanced Topics in Computation Theory OR COMP 630 Formal Semantics of Programming Languages
- COMP 620 Computer System Architecture
- COMP 680 Software Engineering

Computer Science MS

Electives - Choose 4

- Computer Science 400, 500, or 600- level
- Discuss with thesis chair
- 400-level courses must say "approved for graduate credit" in the University Catalog.
- Excluding COMP 450, 480/L, 482, 490/L, 491L, 494, 496ALG, 499, 696, 698 and 699
- At most 2 of the courses can be 400-level

Thesis

- COMP 696C Directed Graduate Research
- COMP 698C Thesis or Graduate Project

Software Engineering MS

Breadth Requirement– All 4

- COMP 582 Requirements Analysis and Specification Formerly 682)
- COMP 583 Software Engineering Management (Formerly 686)
- COMP 680 Advanced Topics in Software Engineering
- COMP 684 Software Architecture and Design

Thesis

- COMP 696C Directed Graduate Research
- COMP 698C Thesis or Graduate Project

Software Engineering MS

SW Engineering Electives - Choose 2

- COMP 584 Advanced Web Engineering
- COMP 585 Graphical User Interfaces
- COMP 586 Object-Oriented Software Development
- COMP 587 Software Verification and Validation
- COMP 589 Software Engineering Metrics

Free Electives- Choose 2

- Computer Science 400, 500, or 600- level
- Discuss with thesis chair
- Excluding COMP 450, 480/L, 482, 490/L, 491L, 494, 496ALG, 499, 696, 698 and 699

Starting Your Thesis Work

1. Find an adviser
2. Complete and submit an R-form
3. Wait for an email with permission number to enroll in COMP 696C
4. Follow adviser's direction
5. Find committee members
6. Complete "planning form" on ETD

Finishing Your Thesis

1. Enroll in COMP 698C, by completing another R-form
2. Submit draft to adviser regularly
3. Submit draft to committee members
4. Submit draft via ETD for formatting review
5. Schedule your defense
6. Complete defense
7. Submit from draft of thesis via ETD

Faculty Research – Selecting an Adviser

http://www.ecs.csun.edu/csgrad/research.html

Service Function Graph Embedding and Routing in Network Function Virtualization

Advisor: Dr. Maryam Jalali (mjalalita@csun.edu)

I. Introduction

Network Function Virtualization (NFV) is an emerging technology that promises to address issues in traditional middleboxes, providing service flexibility and reduced cost. NFV decouples network functions from the proprietary middlebox hardware, thus allowing the network providers to implement network functions on virtual machines running in standard servers. Figure 1 shows an example of the transition from dedicated hardware appliances for network services to software based NFV solutions. Combining NFV with Software-Defined Networking (SDN) technology, future networks such as 5G, mobile networks and optical networks are expected to be operated and utilized at lower cost and higher flexibility.

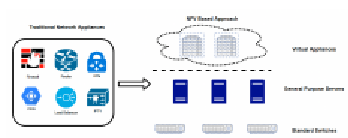

Fig. 1: Traditional to NFV based approach transition

In the NFV environment, an NFV service request is provisioned in the form of a Service Function Graph (SFG) [1]. The SFG, which can be also in form of a chain named Service Function Chain (SFC); defines the exact sequence of actions or Virtual Network Functions (VNFs) that the data stream from the service request is subjected to. These actions or VNFs need to be mapped onto specific physical networks to provide network services for end users. Figure 2 shows an example of an SFG which is a chain of VNFs which consists of network functions Firewall, NAT and a load balancer. To delever an end-to-end service similar to Figure 2, one needs to resolve the SFC Embedding and Routing problem.

Fig. 2: An example of a service function graph

Service Function Chain Embedding and Routing consists of interconnecting a set of network functions through the physical network to ensure network flows are given the correct treatment. These flows must go through end-to-end services traversing a specific set of network functions. This problem can be decomposed into two processes: (i) Embedding, and (ii) Routing. In the embedding process, VNFs are placed onto to virtual machines running on commodity servers. In routing process, the requested functions are connected, which consists of creating physical paths that interconnect the network functions.

Figure 3 shows an example of Embedding and Routing process. The constructed SFC: v1 → v2 → v3, which consists of three VNF nodes, i.e., vi (i = 1; 2; 3). These VNF nodes are connected or chained with VNF links. Each VNF node demands a certain computing resources (e.g., CPU). The available bandwidth of substrate link s1 - s2, s2 - s3, s3 - s4, and s2 - s4 is 15Gbps, 15Gbps, 15Gbps, and 5Gbps, respectively. Accordingly, the constructed SFC can be mapped onto the substrate network to form a service function path s1 → s2 → s3 → s4. To accommodate the NFV service request, VNF nodes v1, v2 and v3 are mapped to substrate nodes (with enough computing resources) s1, s2, and s4, respectively. The VNF links v1 → v2, and v2 → v3 in the constructed SFC are mapped to the physical paths s1 → s2 and s2 → s3 → s4, respectively, as shown by the dash line in Figure 3. Note that substrate link s2 - s4 cannot be used by the request due to the lack of enough bandwidth.

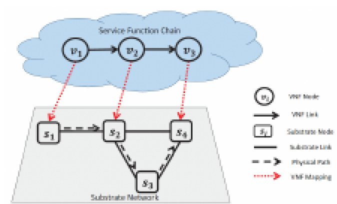

Fig. 3: An example of a Service Function Chain Embedding and Routing

II. Proposed Plan

The example of Embedding and Routing problem in Figure 3 is only for the purpose visualizing the process. In reality, the optimal solution for the problem is much more difficult and belongs to the set of NP-Complete problems. Given this challenging problem, in this project we aim to focus on developing the optimal solution using Integer Linear Programming (ILP) that jointly addresses the embedding and routing problem. For working on the ILP models we will use CPLEX which is an ILP solver software. We will also seek time-efficient heuristic solutions (implemented in Java), which can be adopted for lager size networks. This is a tentative study plan and more details and use case may be added later to the project.

III. Qualifications

As simulations will be done **Java**, **strong programming experience** in Java is required. Interested students must have taken the following courses or their similar versions prior joining the project: **Data Structure and Algorithms, Computer Networks. Interested students must email their transcripts and resumes to (mjalalita@csun.edu).**

References

[1] J. Halpern and C. Pignataro, "Service function chaining (sfc) architecture," Tech. Rep., 2015.

Cool Research Opportunities & Pizza Series (CROPS)

- Short talks is to connect students with faculty doing research
- 3 faculty presentations about their research
- 12:00—1:00pm, Select Fridays (Sept. 9, Sept. 23, Oct. 14, Nov. 4)
- JD 3520 (in-person)
- Pizza provided

Funding

- RA positions available occasionally – talk to faculty members
- Fellowships and Scholarships
  - CECS yearly scholarships
  - Graduate Studies scholarships and funding opportunities
  - University scholarships
- Honors Co-op, TechFest
- Tutoring jobs - http://www.csun.edu/~cecsssc/Tutorial.htm
- Campus student /graduate assistant jobs (including IT) - http://www.csun.edu/usu/jobs/taleo
- Federal Funding Opportunities: https://stemgradstudents.science.gov/
- NSF Graduate Student Fellowship: http://www.nsf.gov/grfp

Admissions Requirements

- **GPA:** Undergraduate grade point average (GPA) of at least 3.0 or a GPA of at least 3.0 from your last 60 units
- **Graduate Record Examination (GRE):** Expect GRE scores in all three sections to be at least or above the 50th percentile
- **Letter of Recommendation:** Optional
- **Statement of Purpose/Resume:** Optional

Program Prerequisites

- Undergraduate coursework for students without BS in Computer Science
- 100-level prerequisites should be completed before applying
- Options:
  - 100-level prerequisite courses offered at community colleges
  http://www.assist.org
  - Open University

Questions

http://www.ecs.csun.edu/csgrad