Research Opportunities in Networking for Undergraduates and MS Students 2015

This page lists several projects in networking, including wireless networking. The projects can generally be scoped to be appropriate for undergraduate or MS students in the Computer Science and Electrical and Computer Engineering departments. Some projects are for-credit research projects (e.g. independent study or honors thesis) but for some projects we are also looking for part-time programmers. In the latter case, you will be paid according to the standard hourly departmental pay rate. Interested students should contact Peter Steenkiste (prs at cs.cmu.edu) and submit a CV/resume and an up-to-date CMU transcript.

**Network privacy and accountability**

In order to balance the need of some users to remain anonymous with need for network operators to be able to hold malicious sender accountable, research presented in a recent paper introduced the concept of delegated accountability. The idea is that senders can use the address of an “accountability delegate” as a source address instead of its own address, thus hiding its identity from observers in the network. Networks, on the other hand, can contact the accountability delegate to check whether traffic is legitimate or to flag suspicious traffic. The paper describes an architecture and a number of possible implementations.

The goal of this project is to implement and evaluate the delegated accountability concept. The first step is to define the necessary protocols and algorithms. The goal is to have an implementation that runs over the IP networks and the XIA future internet architecture.

**Necessary Skills**

- Basic understanding of networks
- C/C++
- socket programming

**Nice-to-have Skills**

- wireshark
- protobuf
- ns (network simulator)
Privacy Leak Detector

User privacy is a major challenge in today’s Internet. The XIA project has been exploring the idea of a “leak detector”, which monitors the user’s traffic and alerts the user if any confidential information is (about to be) sent over the network. We have a basic prototype of leak detector but the back end of the system that monitors the traffic is basic. The goal of this project is to significantly expand the classes of protocols and types of information that the system can work with.

 Necessary Skills

- Basic understanding of networks
- C/C++
- socket programming

 Nice-to-have Skills

- Familiarity with protocols such as HTTP, SMTP, ...
- Monitoring tools such as wireshark

 Hybrid congestion control

The role of congestion control in networks is avoiding that the networks gets overload. Congestion control algorithms fall in two classes. With explicit congestion control algorithms, the network gives explicit feedback to senders about congestion conditions in the network. With implicit algorithms, the sender needs to estimate congestion control conditions based packet drops. These classes have different properties with respect to complexity, efficiency, response time, etc. We have a developed and implemented a hybrid solution that combines features from the two classes. The goal of this project is to implement and evaluate different variants of this algorithm.

 Necessary Skills

- Basic understanding of networks
- C/C++
- socket programming
High-Throughput, Low-Latency, Real-Time Radio Channel Modeling

The CMU Wireless Network Emulator is a high-performance signal processing and simulation system built from a combination of dedicated FPGA-based hardware and general purpose computers. We aim to simulate over 1000 concurrent radio paths, involving over a trillion computations per second and over 100 gigabits per second of digital throughput.

We need you to help us re-architect the general purpose software component to maximize parallelism and minimize latency and jitter. We'll look at multicore processing, concurrent data structures, and modern high performance (userspace / kernel bypass) networking. This project is half software architecture, and half systems hacking.

Necessary Skills

- Systems programming experience (e.g. 15-213, 410, 411, 418, 441)
- Performance-oriented development (e.g. 15-213, 15-418, 18-645)
- C and Java proficiency

Nice-to-have Skills

- Large-project software engineering (e.g. 15-313, 15-405/449/610)
- Low-level hacking experience (e.g. mixed C and assembly, device driver debugging, cache- and multicore-aware development)

Nice-to-know Tools

- Java JNI (Java Native Interface)
- Intel DPDK (Data Plane Development Kit)