NetServ: Extending Click Using Java OSGi Framework

Jae Woo Lee  Suman Srinivasan  Henning Schulzrinne
Dept. of Computer Science, Columbia University, New York, NY, USA
{jae,sumans,hgs}@cs.columbia.edu

This is a talk and demo proposal of an extension to the Click router using the OSGi framework, a Java-based module system. It is based on our experience implementing a prototype of NetServ, an extensible architecture for core network services for the next generation Internet. We will present the design and implementation of the NetServ prototype, followed by a demo showing a simple module being inserted, removed, and upgraded while the Click router is running. We will also present our evaluation results, which indicate that the processing overhead incurred by the Java layer is a reasonable trade-off for the level of modularity we achieve in our system.

NetServ is our on-going research effort to design an extensible architecture for core network services. NetServ defines the virtual services framework, which provides a sandbox-like execution environment for the service modules, offering security, portability, and the ability to control resource allocation. In addition, the framework supports adding and removing service modules at runtime, by network administrators or even by content providers and end users, enabling on-demand and per-flow services in the network core.

We implemented a prototype of NetServ by using the OSGi framework on top of the user-level Click router. Implementing NetServ on the kernel-mode Click is planned as a future work.

OSGi is a component framework for Java. In the OSGi framework, an application is organized as a set of bundles, which can be loaded and unloaded at runtime. This enables installing a new feature into a running application or upgrading a part of it with newly written code.

Figure 1 depicts the overall architecture of the prototype implementation. The shaded boxes represent different components of NetServ, and the thick arrow represents the flow of a packet being forwarded by the router, taking a detour into the NetServ components. The NetServ Click element creates a Java Virtual Machine and launches the OSGi framework, which in turn loads the application bundles.

We will go through the system in detail during the talk, and provide a demo of loading, unloading, and upgrading an application bundle at runtime. We will also present our performance evaluation results.

We would like to disclose that a paper describing the NetServ prototype is currently under review for publication at the ACM ReArch 2009 workshop.