WISL: An Application That Enables User-Perceived Performance Measurement

Richard Liston and Ellen Zegura*

Large scale network measurement at the end-user can guide network architects and managers in activities like choosing application and transport layer parameters, provisioning and routing. They can also help answer such simple, yet elusive questions about the nature of network services provided to typical users. However, the design and implementation of the Internet makes such measurement studies difficult to perform, so studies are often limited in the number and location of points at which measurements are taken. The results of the measurements may not be representative of network performance as experienced by the end-user. Collecting user-perceived performance data requires users to execute measurement tools on their machines. Unfortunately, users have little incentive to participate in performance measurement activities.

To address the incentive problem we present an application called WISL[1], which stands for "What the Internet Sounds Like". The purpose of the application is twofold. At the lowest layer it provides an extensible platform on which to perform and report measurements of end-user perceived network performance. Researchers are able to plug measurement modules into the application with the expectation that the module will be executed at actual end-user locations throughout the world. Each module performs the measurements from its vantage point and the data can be collated for subsequent analysis.

The second, more salient feature of the application is that it plays sounds that are dynamically generated by network events as they occur and are detected by the measurement modules. The primary intent of this aspect of WISL is to provide a platform for composers to create musical compositions that are generated by events occurring on the network worldwide in real time. The music that is generated represents the composer's interpretation of network-level events. It belongs to a class called "aleatoric" music: music generated — at least in part — by chance events. This musical aspect WISL provides the incentive for end-users to execute the application. WISL is intended to be an interesting and compelling musical application in its own right.

We note in addition that the generality of WISL makes it possible to be used as a network monitoring tool. This is accomplished by mapping critical network events to sounds that act as audio alarms.

There are several criteria for the design of WISL. From the perspective of the end-user, WISL must be simple to download and execute. Also, since individual musical tastes vary the end-user must be able to choose among different SoundPalettes — the packages contributed by composers. Choosing a SoundPalette can be compared to selecting a particular radio station. There can be many different SoundPalettes for the user to choose from. Meeting these criteria will increase the likelihood of participation by end-users and therefore the quality of data collected.

Network researchers must be able to easily incorporate NetModules — the modules that perform the measurements. WISL must also allows for performing a wide range of types of measurements, and be able to accurately timestamp events. Researchers must be able to log data for subsequent analysis. These criteria will ensure that WISL will provide the most useful platform for network researchers. They will also help to attract network researchers to contribute NetModules that report on many different kinds of network events.

WISL must offer the composer the ability to create a representation in sound of a wide variety of events that may occur on the network. A rich source of network event detection will make for a varied and interesting platform upon which composers can create highly varied sound environments. It must also allow the composer to select and combine the events in different ways and at different time scales. The composer must not be bound to using events only at the times that the NetModule creator specifies.

For portability, WISL is implemented in Java and utilizes the JavaSound API. Although Java can be restrictive for some types of network measurement, we have been successful in implementing several interesting, as well as accurate, "atomic" NetModules. We also have created a complex NetModule called LandmarksModule. LandmarksModule collects data to use in evaluating different schemes for network topology mapping. LandmarksModule works by querying a fixed set of DNS servers — the thirteen gTLD servers — as network landmarks. It constructs a vector of RTTs and sends this to a central server which decides what other currently running WISL nodes are topologically close neighbors and sends this list to the WISL node.

Our primary goals in this poster session are to introduce the WISL application to the research community; to outline our criteria for WISL and to present its design in detail; and to discuss the kinds of measurements that WISL would be well-adapted for, as well as its limitations.

*Richard Liston is with Ursinus College and Ellen Zegura is with Georgia Tech, Email: liston@ursinus.edu, ewz@cc.gatech.edu

[1]For more information and download visit www.wisl.info