# Automatic Service Selection in Dynamic Wireless Network Environments


George Lee Peyman Faratin Steven Bauer John Wroclawski
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
{gjl,peyman,bauer,jtw}@lcs.mit.edu


## 1 Introduction

Our poster considers the use of machine learning agents to autonomously and continually select among wireless access services available to a user. Our context is the Personal Router project; a technical research program aimed at reshaping the wireless network access market towards greater competition and diversity [CW00]. By allowing potential suppliers to easily advertise service offerings in a given area, and allowing users to transparently and continually discover, negotiate terms for, and switch between available network services, we aim to greatly lower the barriers to entry in the wireless service market, creating a rich ecosystem of large and small service providers with different offerings, capabilities, and business models.

A critical function within the PR vision is that of *service selection*. The Personal Router, acting on behalf of its user, must intelligently, intuitively, and autonomously select from among the available services the one that best meets its user's needs. It must do this transparently, without involving, bothering, or distracting its user. Its task is difficult – acceptable service choice depends not only on the features and cost of the service, but also on the context of the situation, including such dynamic variables as the applications the user is running and the user's higher level goals. Beyond this, the set of available services may change rapidly with time and the user's location, requiring the PR to choose new services frequently. Without this automatic selection capability, users could not possibly respond to the constantly changing set of services and application demands.

Previous approaches to wireless service selection, including static policies and manual selection, are inadequate. Static policies cannot accommodate individual user preferences, and users are reluctant to constantly interact with a user interface for manual selection, particularly in rapidly changing and complicated wireless service environments.

These considerations motivate us to explore a machine learning approach, in which an intelligent agent learns user preferences and makes selections automatically on behalf of the user with minimal user involvement. Our poster presents this approach in three parts. First, we give an overview of the challenges, complexity, and importance of the service selection problem and the assumptions we make about the network and user. Next we present an architecture for service selection in the PR and describe its components. Finally we illustrate the performance of our system in dynamic and partially unobservable network environments.

## 2 Network Services

Service providers may offer a wide variety of network services. We characterize these services in terms of *service profiles*. We choose to describe services in terms of a two token bucket profile [Yan02], price per second, and price per byte. The two bucket profile consists of an average data rate, a burst size, and a refill interval, enabling it to more accurately capture the quality characteristics of a service that matter to the user. Service profiles can describe high and low bandwidth services, burstiness over different time scales, as well as different pricing plans.

The work described in this poster assumes the availability of accurate service profiles and that the PR has a way of discovering the set of services available. These capabilities are implemented by other modules in the PR system.

## 3 System Architecture

The PR agent comprises four components to address the problem of service selection: a *service evaluator* that learns how the user perceives the service currently in use, a *service change controller* that decides when to switch services and which one to select, a *service value predictor* to estimate the value to the user of services not yet experienced by the agent, and a *user interface* that presents information to the user and passes user feedback to the other components. Together, these components allow the PR to learn the value of services from user feedback, adapt to changing user needs, and estimate the value of new services [FLWP03].

### 3.1 User Interface

For the user interface (UI) to be effective, it must be intuitive and unobtrusive. At a minimum, the user must be able to provide feedback about their satisfaction with the current service and to indicate whether they desire a higher quality or a lower cost service. In addition, to help guide the PR's decisions we also allow them to indicate their tolerance for trying new services. Thus the UI presents the user with four buttons: *better*, indicating dissatisfaction with the quality of the current service and requesting a higher quality service; *cheaper*, expressing dissatisfaction with cost and requesting a lower cost service; *explore more*; and *explore less*. From these button presses, the UI needs to extract four values to use as input into the other components: $\Delta q$, $\Delta c$, $\Delta w$, and $\Delta x$, the amount to change the quality estimate, cost estimate, the weighting between quality and cost, and exploration level, respectively. The user interface also informs the user of the cost and quality of the current service.

### 3.2 Service Evaluator

The agent's selections should ultimately be based on user preferences, so we choose to evaluate services in terms of user perceived cost and quality. We take a reinforcement learning based approach, updating quality and cost estimates based on user feedback, allowing the PR to learn individual user preferences without depending on service advertisements. Since user perceptions of cost and quality may depend on features of the user context such as the current application, the PR maintains separate estimates for each user context, averaging the $\Delta q$ and $\Delta c$ values from the UI using an exponentially weighted moving average.

### 3.3 Change Controller

The change controller decides when to switch services and which services to select based on information from the service evaluator and from the user interface. It combines the quality and cost estimates from the service evaluator linearly using the $\Delta w$ values from the UI to compute a utility $u$. The change controller then selects a service stochastically based on a Gibbs softmax distribution of the estimated utility of the available services, where the probability of selecting a service $s_i$ with utility $u(s)$ from the set of available services $S$ is given by the expression

$$\frac{e^{u(s)/\tau}}{\sum_{x \in S} e^{u(x)/\tau}}$$

The exploration level of the service evaluator and the user's willingness to explore affect the the temperature parameter $\tau$, changing the frequency with which the PR randomly switches services. We chose this type of selection policy because it can effectively balance the need to explore untried services with the desire to select the highest utility services.

### 3.4 Service Value Predictor

Using the evaluator allows the agent to learn the value of individual service profiles, but does not help it select from services the user has not yet experienced. To improve selections when new services are encountered, the PR forms a model of user utility to predict the value of new services based on previous user feedback and the current service profile. Of the many possible approaches to the prediction problem, including linear regression and interpolation, we choose to use multilayer neural networks (NN) because they can learn to approximate any function given enough training data. The PR maintains a separate NN for each user context, one for quality and one for cost. These NN learn a mapping from service profiles to estimated quality and cost based on the $\Delta q$ and $\Delta c$ from the UI.

## 4 Results and Evaluation

We implemented a prototype of the PR in software and have evaluated it using simulations, showing that our approach can effectively learn a model of user preferences even when new service profiles are encountered and when service descriptions are incomplete. We have also begun to test our system with real users to evaluate its performance in more realistic scenarios.

## References


[CW00] D.D Clark and J. Wroclawski. The personal router whitepaper. *MIT Technical Report*, 2000.

[FLWP03] P. Faratin, G. Lee, J. Wroclawski, and S. Parsons. Social user agents for dynamic access to wireless networks. In *Proceedings of AAAI Spring Symposium on Human Interaction with Autonomous Systems in Complex Environments*, Stanford, CA, US, 2003.

[Yan02] Xiaowei Yang. Designing traffic profiles for bursty internet traffic. In *Proceedings of IEEE Global Internet*, Taipei, Taiwan, 2002.