Wireless Networks 8, 213–230, 2002

An Architecture for Secure Wide-Area Service Discovery

TODD D. HODES, STEVEN E. CZERWINSKI, BEN Y. ZHAO, ANTHONY D. JOSEPH and RANDY H. KATZ
Computer Science Division, University of California, Berkeley, USA

Abstract. The widespread deployment of inexpensive communications technology, computational resources in the networking infrastructure, and network-enabled end devices poses an interesting problem for end users: how to locate a particular network service or device out of hundreds of thousands of accessible services and devices. This paper presents the architecture and implementation of a secure wide-area Service Discovery Service (SDS). Service providers use the SDS to advertise descriptions of available or already running services, while clients use the SDS to compose complex queries for locating these services. Service descriptions and queries use the eXtensible Markup Language (XML) to encode such factors as cost, performance, location, and device- or service-specific capabilities. The SDS provides a fault-tolerant, incrementally scalable service for locating services in the wide-area. Security is a core component of the SDS: communications are both encrypted and authenticated where necessary, and the system uses a hybrid access control list and capability system to control access to service information. Wide-area query routing is also a core component of the SDS: all information in the system is potentially reachable by all clients.

Keywords: network protocols, service discovery, location services, name lookup

1. Introduction

The decreasing cost of networking technology and network-enabled devices is enabling the large-scale deployment of both [51]. Simultaneously, significant computational resources are being deployed within the network infrastructure, and this computational infrastructure is being used to offer many new and innovative services to users of these network-enabled devices. We define such "services" as applications with well-known interfaces that perform computation or actions on behalf of users. For example, an application that allows a user to control the lights in a room [23] is a service. Other examples of services are printers, fax machines, music servers, and web services such as the FreeDB.org CD database.

Ultimately, we expect that, just as there are hundreds of thousands of web servers, there will be at least hundreds of thousands of services available to end users. Given this assumption, a key challenge for these end users will be *locating* the appropriate service for a given task, where "appropriate" has a user-specific definition (e.g., cost, location, accessibility, etc.). Clients cannot be expected to track which services are running or to know which ones can be trusted. Thus, clients will require a directory service that enables them to locate the services that they are interested in using, and this service will have to address such issues as trustworthiness, secure access, (dis)trust management, endpoint mobility, complex query support, and scaling behavior. We have built such a platform, the Ninja[1] *Service Discovery Service* (SDS). The SDS enables clients to more effectively search for and use the services available via the network.

[1] The Ninja project is developing a scalable, fault-tolerant, distributed, composable services platform [19].

The SDS is a scalable, fault-tolerant, and secure information repository providing clients with directory-style access to all available services. The SDS can store many types of information, including descriptions of services that are available for execution ("unpinned" services), services running at specific hosts ("pinned" services), available service platforms, and passive data. The SDS supports both push-based and pull-based access; the former allows passive discovery, while the latter permits the use of a query model.

Service descriptions and queries are specified in eXtensible Markup Language (XML) [4], leveraging the flexibility and semantic-rich content of this self-describing syntax.

The SDS also plays an important role in helping clients determine the trustworthiness of services, and vice versa. This role is critical in an open environment, where there are many opportunities for misuse, both from fraudulent services and misbehaving clients. To address security concerns, the SDS controls the set of agents that have the ability to discover services, allowing capability-based access control, i.e., to hide the *existence* of services rather than disallowing access to a located service.

As a globally-distributed, wide-area service, the SDS architecture addresses challenges beyond those that operate solely in the local area: network partitions, component failures, potential bandwidth limitations between entities, workload distribution, and application-level query routing between components.

This paper presents the design of the SDS, focusing on the architecture of the directory service, the security features of the system, and the wide-area query model. Section 2 describes the system design concepts. Section 3 discusses the SDS architecture and its security features. Section 4 discusses wide-area operation. Section 5 presents performance measurements from the SDS prototype implementation. Section 6