

# 既存のネットワークサービスに サービス発見機能を組み込む手法の設計と実装

小川 貴央†      下川 俊彦‡      牛島 和夫‡

†九州大学大学院システム情報科学研究科      ‡九州大学大学院システム情報科学研究院

## 1 はじめに

近年、携帯型計算機の普及により、動的にネットワークに接続できる計算機が増加している。そのような計算機において、最寄りのプリンタから出力したり、最適なweb キャッシュを利用したりするには、設定の変更も動的に行う必要がある。また、ネットワークサービスの利用に必要な設定は複雑であり、かつ、設定作業は一般的に手動である。そのため、利用者の負担となっている。よって、自動設定機能の必要性が高まっている。

自動設定を行うためには、次の三つの段階が必要である。

1. ホスト名やプリンタの設置場所のような、利用に必要なサービス情報を入手
2. 入手したサービス情報を評価
3. サービス情報を設定に反映

サービス情報の入手方法には、SLP(Service Location Protocol)<sup>[6]</sup>, Jini<sup>[2]</sup>, SDS (Service Discovery Service)<sup>[3]</sup>, INS (Intentional Naming System)<sup>[4]</sup>といったサービス発見機構がある。SLPでは図1のように、サーバ側がサービス情報を広報し、ネットワーク上の収集サーバがサービス情報を蓄積する。そして、クライアント側がサービス情報を検索する機能によって必要とするサービス情報の入手を可能にしている。

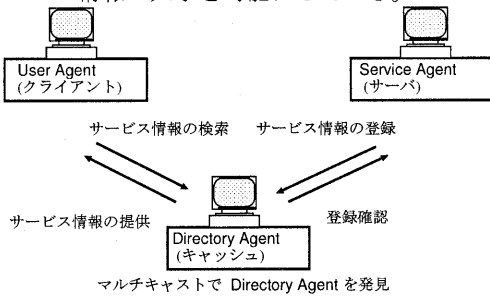


図1 サービス発見機構 (SLP)

既存のサービスがこれらのサービス発見機能を利用し自動設定を行うためには、サーバ、クライアント両方にサービス発見機構と通信する機能を組み込む必要がある。さらにクライアントは、サービス情報の評価、設定変更を行う機能が必要である。また、サーバは、サービス発見機能で広報するためのサービス情報を作成しなければならない。これらに対応する労力が高いため、サービス発見機能が利用されていない。また、クライアントはサー

ビス発見機能に対応したサーバがないため対応が進まず、サーバもクライアントがないため対応が進んでいない。

本研究では、既存のサービスにサービス発見機能を円滑に組み込み、自動設定機能の普及を支援する事を目的とする。そのために、既存のサービスとサービス発見機能との連携を円滑に行うシステムの開発を行う。

## 2 既存のサービスにサービス発見機能を組み込む手法

従来のサーバ、クライアントが円滑にサービス発見機構を利用するために、以下の機能を図2のように独立に実現する。

### クライアント側支援機能

検索条件を作成しサービス発見機構からサービス情報を入手する。得られたサービス情報を評価し、設定に反映する。

### サーバ側支援機能

サーバからサービス情報を取り出しサービス発見機構に通知する。

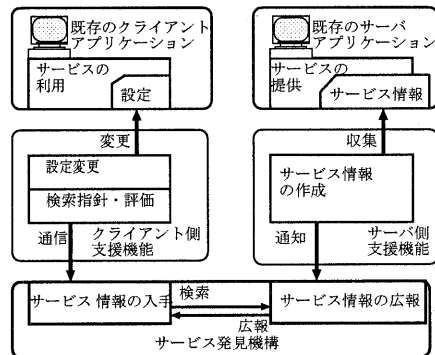


図2 既存のサービスにサービス発見機能を組み込む手法の概要

このような構成にすることで、既存のサーバ、クライアント及び、サービス発見機構に変更を加える事なく自動設定機能が利用できる。クライアント側支援機能によって、自動設定が可能なクライアントを増やし、サーバ側支援機能によって、自動設定の対象となるサーバを増やす。

サーバ側支援機能のサービスを取り出す方法には次の2つの方法がある。

### 直接的にサービス情報を収集する方法

ネットワークを通じて情報を収集、広報する。

### 間接的にサービス情報を収集する方法

既存サーバの設定ファイルから情報を収集、広報する。

直接的にサービス情報を収集する方法は、定期的な情報を集める事で、動的なサーバの追加、削除に対応で

Design and Implementation of Embedding Service Location Function into Existing Network Services.

Takahiro Ogawa, Toshihiko Shimokawa, and Kazuo Ushijima.

Graduate School of Information Science and Electrical Engineering, Kyushu University.

きる。また、一カ所で動作させる事でネットワーク上の全てのサービス情報を収集できる。しかし、サーバがサービスの広報について、可否を制御できない。

間接的にサービス情報を収集する方法は、ネットワークから取得できない情報の取得も可能である。また、既存のサービスの多くはファイルに設定情報を持っているので適用しやすい。しかし、設定ファイルを指定する手間がかかり、また、設定ファイルは実装毎に異なるため、個々の実装毎に対応する必要ある。

### 3 実装

サービス発見機構に SLP を利用し、既存のサービスとして Unix で標準的に使われている lpr (Line Printer Daemon Protocol) [4] 及び、Windows で標準的に使われている SMB(Server Message Block) [3] によるプリンタサービスに適用した。開発言語には Ruby<sup>[7]</sup> を用いた。

#### 3.1 クライアント側支援機能の実装

サーバの選択方法は、自動的に判断するのは難しいため、利用者の判断によって選択する方法を採用した。そのための入出力機能を実装した。自動的な判断が難しい理由は、サーバ側で提供するサービス情報が限られているためである。また、プリンタの選択では位置情報が重要だが、計算機上で物理的な位置情報を自動的に取得するのが困難なためである。

既存のクライアントの設定を変更する方法は次の2つを実装した。

1. 既存のクライアントを呼び出す方法

2. 既存のクライアントの設定ファイルを書換える方法。

1. は SMB、lpr プリンタに適用した。サービスの利用毎にサービス情報を収集し、既存のクライアントを呼び出す。設定は引数として与える。2. は lpr プリンタに適用した。収集したサービス情報をもとに設定ファイルを変更する。設定ファイルを変更できる権限が必要である。

#### 3.2 サーバ側支援機能の実装

直接的にサービス情報を収集する方法は、SMB プリンタに適用した。SMB を用いて、ネットワーク上にある計算機からプリンタサービスを発見し、SLP を用いて広報する。収集可能なサービス情報は、サーバのアドレス、プリンタ名、管理者による注釈である。

間接的にサービス情報を収集する方法は、lpr プリンタに適用した。lpr プリンタの設定ファイルを読み込み、サーバのアドレス、プリンタ名を広報する。

現状の問題として、サーバ側支援機能では得られるサービス情報は限られていることが挙げられる。しかし、自動設定可能なクライアントが増加すれば、自動設定に対応するサーバが増加し、十分なサービス情報が得られるようになると予想する。

### 4 評価

クライアントがサービス情報を入手するのに要する時間を計測した。サービス情報を蓄積するサーバに登録された500のサービスの中から、プリンタサービスを探索し、結果として5つのプリンタの情報を入手した。クライアントには、11Mbpsの無線接続、CPU Celeron

500 Mhz の計算機を用い、サービス情報を蓄積するサーバには 100 Mbps イーサネット接続、CPU Pentium Pro 200 Mhz の計算機を用いた。この試行を 100 回行い平均を求めた。クライアントが要求を出して結果を受け取るまで、1 回あたりの平均時間は、0.055 秒になり十分小さい。また、この際のサービス情報を取得するのに通信量は 1 サービスあたり 1820 byte と十分小さい。

### 5 おわりに

既存のサービスにサービス発見機能を組み込む手法に関する設計、実装を行った。また、クライアントの設定に要する時間の評価を行い、十分実用的な時間で設定を変更できる事を示した。

今後、サーバ側の評価を行う予定である。評価としては、サーバ側支援機能のネットワーク負荷を検討している。収集できるサービス情報についての評価も行う。また、クライアントのサービス情報の評価方法の検討、他のサービスへの適用も行う予定である。

### 参考文献

- [1] Erik Guttman, Sun Microsystems, "Service Location Protocol: Automatic Discovery of IP Network Services," IEEE Internet Computing 3(4), pp. 71-80, July/August 1999
- [2] Sun Microsystems, "Jini Technology Executive Overview Protocol," Jan. 1999;  
<http://www.sun.com/jini/overview/overview.ps>
- [3] Steven E. Czerwinski, Ben Y. Zhao, Todd D. Hodes, Anthony D. Joseph, and Randy H. Katz, "An Architecture for a Secure Service Discovery Service," Computer Science Division University of California, Berkeley Proc. ACM/IEEE MOBICOM, pages 24-35, August 1999.
- [4] William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, and Jeremy Lilley, "The design and implementation of an intentional naming system," 17th ACM Symposium on Operating Systems Principles (SOSP '99) Published as Operating Systems Review, 34 (5):186-201, Dec. 1999
- [5] Microsoft Networks, "SMB FILE SHARING PROTOCOL EXTENSIONS SMB File Sharing Protocol Extensions Version 3.0 Document Version 1.09," November 29, 1989;  
<ftp://ftp.microsoft.com/developr/drg/CIFS/smb.txt>
- [6] L. McLaughlin III, "Line Printer Daemon Protocol," RFC1179, August 1990;  
<ftp://ftp.ietf.org/rfc/rfc1179.txt>
- [7] まつもとゆきひろ, 石塚圭樹, "オブジェクト指向スクリプト言語 Ruby," アスキー出版局