Design, Development and Implementation of a Registration Server for the NetTraveler Middleware System

José F. Enseñat Acevedo
Advisor: Manuel Rodríguez Martínez
Electrical and Computer Engineering Department
University of Puerto Rico, Mayagüez Campus
Mayagüez, Puerto Rico 00681-5000
jensenat@ece.uprm.edu

Abstract

The next wave of information systems will feature data sources residing on handheld devices, laptops computers, embedded systems, sensors and other types of small-sized devices. The mobility advantage of these devices causes a new set of problems not seen in common networks. Lost of connectivity, low bandwidth and limited power sources are some of the new challenges presented to developers working in this area. We are in the process to create a Registration Server for NetTraveler: a database middleware system to integrate these mobile devices as reliable data sources in a computer network. This paper describes this Registration Server and how it works.

1. Introduction

Middleware systems provide some advantages in a distributed environment [Finkelstein95]. Moreover database middleware systems were developed to integrate heterogeneous collections of data sources distributed over a network as a homogeneous unit [Deutsch94]. Typically data sources reside on computers or servers that usually are online on a fixed location with continue power source. With the emergence of broadband network, wireless networks and computer technology for handheld devices this assumption has become less certain. The next wave of information systems will feature data sources residing on handheld devices, laptops computers, embedded systems, sensors and other types of small-sized devices. Mobile devices have increases dramatically its computational powers in recent years. Mobile phones and PDAs are the best examples but even there are hand watches running fully feature operating systems [Narayanaswani02]. The mobility advantage of these devices causes a new set of problems not seen in common networks. Lost of connectivity, low bandwidth and limited power sources are some of the new challenges presented to developers working in this area. To efficiently incorporate these low capacity [Li01] devices and their data sets it will be necessary to integrate them into coherent information systems, which must also incorporate the more traditional data sources (e.g. relational databases).

In order to effectively integrate these new networks technologies we are developing NetTraveler, a database middleware system to integrate data sources residing on PDAs, mobile laptops, embedded devices and enterprise servers. NetTraveler is designed to cope with
dynamic wide-area environments where data sources go off-line, change location, have limited power capabilities, and form ad-hoc federations of sites that work together to complete a given task and then go about their business independently.

In this paper we will discuss an implementation of a Registration Server for NetTraveler. The Registration Server will be used to coordinate access to a federation and to the resources that it provides. In addition the Registration Server handles all bookkeeping necessary to log events such a departure of a federation member, or a reappearance of a member at a different network. Registration Servers located at cooperative local execution environments will work together to keep track of moving data sources and route data request as appropriate.

2. Registration Server Overview

The Registration Server will be the entrance point to a NetTraveler federation. Once a member validation is complete, access to the federation is granted. There are different types of federations which rely on different types of validations. Also members for federations may have different levels of permissions. Members can belong to more than one federation and may have different permission levels on each federation. Members can login from any local execution environment in the cooperative network, gaining access to the same services and data that he had before provided these services and data are still available. Moreover this device should still be able to serve data and resources to the clients that were previously accessing it. Thus, with NetTraveler users should be able to have their data “follow” them as they move. Notice that this is different from the problem of routing addressed by Mobile IP, where routers cooperate to forward packages as a host changes networks. NetTraveler is addressing problem at the application layer. It allows devices to access new resources and data that become available at their new environment, while keeping access to the resources and data in the old environment (see Figure 1 and Figure 2).

![Figure 1. Local execution environment with 3 members sharing resources among them.](image1)

![Figure 2. The PDA moved to local execution environment #2 but still sharing resources with members from local execution environment #1](image2)

3. Federations

Federations are the fundamental organizational unit in NetTraveler. An ad-hoc federation of peer sites is a collection of computing devices, including the mobile ones, which have agreed or are
configured to work together as a group to exchange data and share computational resources. A federation can span one or more local execution environments. Execution environments can be compose of a combination of 10/100/1000 Ethernet, and wireless networks (i.e. IEEE 802.11b). Each local execution environment will have it own Registration Server.

3.1 Federations types

There are three classes of federations defined in NetTraveler. The three types of Federations are:

- **Public Federations** – These are federations in which any device connected can get access. They are similar to file sharing systems like Gnutella [B]

- **Private Federations** – These are federations that only accept certain resources. Typically, some arrangement and security clearance will be required to join a private federation.

- **Associative Federation** – In this case access to a federation is granted by the majority of the members. This type of federation can be used when Quality of Service (QoS) guarantees are to be maintained. If the arrival of a new member poses too burden to the system, the members can vote to deny access and maintain the current level of service.

3.2 Federation maintenance

The Registration Server must provide a series of protocols to form, maintain and dissolve ad-hoc federations according to the type of federation we are managing.

- **Federation Formation** – This operation is used to establish a new federation. A federation must contain at least a member to be considered active.

- **Federation Dissolution** – In this case all the metadata about the federation will be moved to a repository for archival purposes. In addition Registration Servers in cooperation environments will exchange information to notice host sources about the demise of the federation.

4. Members

A member can be any device requesting or offering data sets or services. Members can connect to a federation through the Registration Server. After a successful connection they can share or request data sets and resources according to the types of services the member can enjoy.

4.1 Sharing data and resources

A member can share their data and resources through the Registration Server. In public federations any member can subscribe a resource or data set to make it available to everyone on the federation. Private federations will be defined to accept only some types of data or resources, members not following that restriction can be banned from the federation by the Registration Server.

4.2 Requesting data and resources

Once the resources subscription has finished, others members can have access to them sending the appropriate request to the Registration Server. The server returns the necessary information to establish a direct connection with the member offering the resource.
4.3 Moving across networks

If a member has to disconnect from a local execution environment one of two states will be set. If it is an expected disconnection, when the data source disassociate it will indicate if the action is temporary or permanent and the appropriate state information will be generate and store by the Registration Server. If the disconnection is unexpected then the Registration Server, after a predefined time, will generate a permanent disconnection state. If disconnection is temporary the member re-association will allow old peers to know about the return of the resource and pending operations will be re-started. If disconnection is permanent the system will help recover query work that cannot be finished because of the departure of the resource and will store information that can be used to restart an operation should the resource return. Registration Servers located at cooperative local execution environments will work together to keep track of moving data sources and route data request as appropriate.

5. Conclusions

Mobile devices always have been considered from the client side. Due to the many limitations presented by these devices we still can not take them as reliable data sources. Despite this in the two last years this new technology has made a huge improvement and while there still some limitations, there are some capabilities and advantages that we can use in our favor.

References


[Li01] – Li, Jinyang; Blake, Charles; De Couto, Douglas S.J.; Lee, Hu Imm; Morris, Robert "Capacity of Ad Hoc Wireless Networks" SIGMOBILE’01, (July 2001) Pages 66-69