Transparent Resource Discovery for Mobile Computers*

Pravin Bhagwat
Computer Science Department
University of Maryland
College Park, MD 20742

Charles E. Perkins
T.J. Watson Research Center
IBM
Hawthorne, NY 10562

Satish K. Tripathi
Computer Science Department
University of Maryland
College Park, MD 20742

Abstract

As mobile computers move from one place to another their "physical" points of attachment to network change from time to time. Host movement causes service disruption. Mobile-IP protocols attempts to solve this problem by hiding the effect of change in network service access point from the transport and higher layers. Thus, Mobile-IP ensures that a mobile host "virtually" remains connected to its home regardless of its current point of attachment. This allows existing applications to operate over mobile nodes without any modifications. While away from home, mobile hosts continue to access services from servers located at the home network. Applications running on mobile hosts, therefore, experience increased latency as accesses to home resources (such as name server, file server etc.) are routed over multiple links and routers. To alleviate this problem, new mechanisms are required that enable mobile hosts to discover and access needed resources on the foreign network. In this paper, we outline alternative approaches for providing such mechanisms and discuss their relative merits and limitations.

The recent advances in wireless communication technology, coupled with the progress in hardware miniaturization, are taking us into a new era of computing, popularly known as nomadic computing. Battery powered, untethered computers are likely to become a pervasive part of our computing infrastructure. There are, however, many technical challenges involved before the vision of ubiquitous computing can be realized. Paramount among these is the challenge of providing continuous, location independent network access to mobile computers. Traditionally, the data networking protocols have been designed based on the assumption that network nodes are stationary. With the emergence of laptop computers this assumption no longer holds. Users of laptop computers want to carry their computers with them and experience the same computational environment regardless of their geographical location. From the network providers point of view, this constraint gives rise to many tricky problems in the areas of routing, address resolution and resource discovery.

Over the last two years, many enhancements to existing networking protocols have been suggested which enable mobile hosts to move around a network [2, 4, 5, 6]. Ideally, mobile hosts can change their network attachment point, in a fashion which is completely transparent to the transport layer protocols and the applications running above them. All proposed solutions for Mobile-IP, for example, meet this goal. The Mobile-IP approach [3] automatically provides *location transparency*. Portable Computers, like any other desktop computer, need to access resources such as NFS file system, name servers, etc. Existing portable systems are configured to make use of resources available on their home networks. As a result, portable systems continue to access services from the home network even though they might not be physically connected to it. This restriction gives rise to many performance problems since every access to home servers is routed across multiple, possibly slow, links. Slow links introduce delays and cause performance degradation visible to the mobile user. By developing a *resource discovery protocol*, we can hope to allow more local access to needed resources, and thus provide *performance transparency* to mobile users in addition to *location transparency*.

*This work is supported in part by NSF grant CCR 9318933

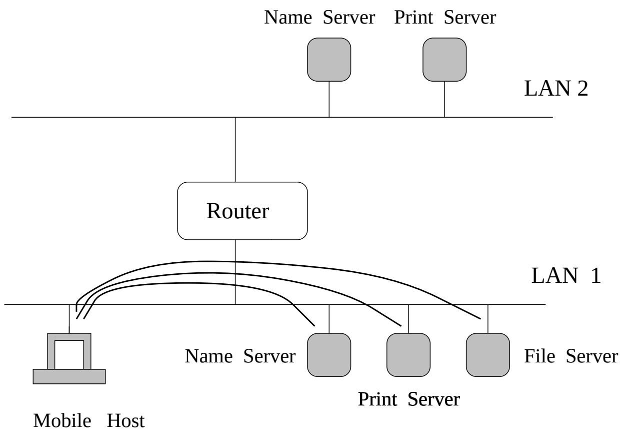

Figure 1: A Mobile host connected to its home network

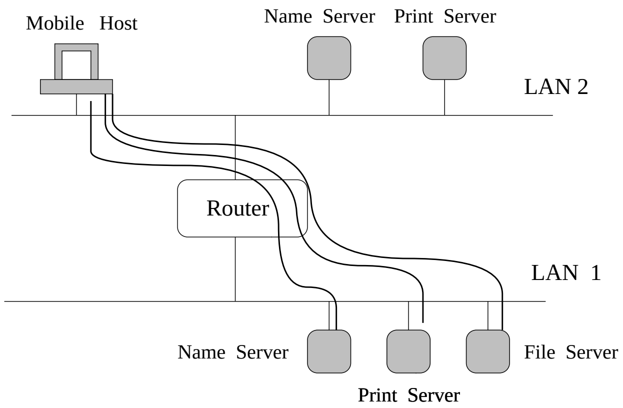

Figure 2: A Mobile host away from its home network

Figure 1 shows an example of a typical client-server interaction which is common to all UNIX, OS/2 LAN installations. Within LAN environments, the client-server paradigm is very widely used for services such as NFS, Yellow Pages, Name Servers, Print Servers etc. The client-server paradigm does not impose any restrictions on the location of client and servers. However, for performance reasons it is a common practice to locate clients and servers on the same LAN. This method of allocation ensures that the majority of traffic does not cross LAN boundaries, thereby reducing the load on routers and bridges. Limiting the client-server interaction traffic within LAN premises also minimizes the chance of a packet loss.

Host mobility violates the locality of reference assumption within the aforementioned environment. Figure 2 shows how access to home resources translates into extra overhead on routers and networks.

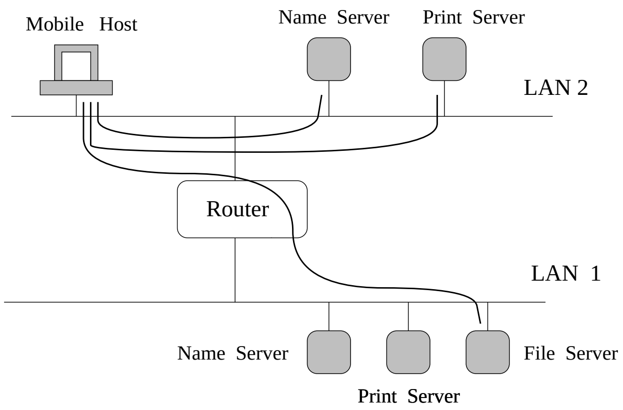

Figure 3: Local Resource Discovery

When away from home, the use of locally available resources within the foreign network is crucial for achieving *performance transparency*. Our objective is to design and integrate a *resource discovery protocol(RDP)* within UNIX systems. We would like such a resource discovery protocol to operate in a fashion completely transparent to the client-server applications so that they continue to run without any modifications.

Resource discovery packets could be targeted towards servers which maintain availability information for many classes of resources. We expect that the address of such a resource location protocol server would be a popular item of information to be delivered by local DHCP [1] servers to new clients.

To achieve the maximum benefit, the resource discovery protocol should be invoked each time a mobile host crosses a network boundary. We propose to add a new signal called `SIG_CELLSWITCH` which would be invoked on behalf of each client application that wishes to connect to a local server. RDP can be tied with the signal handler so that it is invoked each time a network boundary is crossed by the mobile host. After local resources are discovered, connections to previous servers can be suspended. Each LAN installation is not guaranteed to have every type of server available. In such cases, the mobile host can continue to access services from the home location. Figure 3 shows an example of a mobile host which is equipped with RDP.

Many applications may be able to take special action when a cell switch occurs. For this reason, we would like to investigate what is possible when a signal handler can be integrated with such applications at the user code level. We may also be able to provide benefits to applications without changing source code, just by relinking them to special mobility-aware libraries. Thus, there are three possible ways to help mobile applications become better able to handle the requirements of cell switches:

Approach	Application Changes
Kernel level support	None required
Source code compatible user library support	Only relinking required
New Application Programming Interface	Code revision required

Kernel Level Support Servers are always accessed via well defined port numbers. This feature can be effectively made use of by designing necessary kernel level support that causes socket accesses to be redirected to a local resource provider. We only need to maintain a mapping between port numbers and the addresses of corresponding servers which are locally available. This mapping should be updated by invoking RDP protocol each time a cell switch occurs.

Mobility aware Library support Replacing existing libraries with libraries that include signal handlers for `SIG_CELLSWITCH` may provide valuable assistance to applications without any requirements for source code changes. Existing applications only need to be relinked with mobility aware libraries.

New API Maximum gains can only be expected by making future applications mobility-aware. To this end, we expect to define a new applications programming interface for easy manipulation of the `SIG_CELLSWITCH` signal. With slight modifications existing applications too can take advantage of this new feature to become more mobility-aware.

The gains which are to be expected from the use of a resource discovery protocol are summarized below:

- Faster and low latency access to network resources
- Reduced load on routers and networks
- Reduced static configuration, allowing easier and often automatic reconfiguration even for non-mobile office environments
- Improved fault tolerance. Server failure at the home network does not affect the operation of mobile hosts which happen to be connected to foreign networks.
- Coexistence with Mobile-IP, providing additional features
- Operates by adding new services, not modifying existing infrastructure
- Processing requirements and network load are minimal

The applicability of the system described herein is not limited to scientific computing environment alone. Many applications can be envisioned where the proposed mechanisms can be effectively used. For example, we can envision a setup consisting of handheld devices capable of displaying video clips and a backbone network with video servers attached at various points. As mobile hosts move from the range of one base station to another, they also switch from one video server to another if such a switch is deemed necessary to reduce load on the network. Decision to switch could also depend on some policy constraints and QOS parameters.

References

[1] R. Droms. Dynamic Host Configuration Protocol. RFC 1541, October 1993.

[2] John Ioannidis, Dan Duchamp, and Gerald Q. Maguire Jr. IP-based Protocols for Mobile Internetworking. In *Proceedings of ACM SIGCOMM*, pages 235–245, 1991.

[3] Charles Perkins. draft-ietf-mobileip-protocol-07.txt. Draft RFC - work in progress, Nov 1994.

[4] Charles Perkins and Pravin Bhagwat. A Mobile Networking System Based on Internet Protoccol. *IEEE Personal Communication Magazine*, 1(1), Feb 1994.

[5] Fumio Teraoka and Mario Tokoro. Host Migration Transparency in IP Networks. *Computer Communication Review*, pages 45–65, Jan 1993.

[6] Hiromi Wada, Takashi Yozawa, Tatsuya Ohnishi, and Yasunori Tanaka. Mobile Computing Environment Based on Internet Pack et Forwarding. In *proceeding of Winter USENIX*, pages 503–517, San Diego, CA, Jan 1993.