The Migrate Approach to Internet Mobility[1]

Alex C. Snoeren SNOEREN@LCS.MIT.EDU
Hari Balakrishnan HARI@LCS.MIT.EDU
M. Frans Kaashoek KAASHOEK@LCS.MIT.EDU
MIT Lab for Computer Science, 545 Technology Square, Cambridge, MA 02139

1. Introduction

The proliferation of laptops, handheld computers, cellular phones, and other mobile computing platforms connected to the Internet has triggered much research into system support for mobile networking over the past few years. Yet, when viewed as a large-scale, heterogeneous, distributed system, the Internet is notoriously lacking in any form of general support for mobile operation.

We argue that previous work has failed to comprehensively address several important issues. We are developing a session-oriented architecture called *Migrate* that preserves end-to-end application-layer connectivity by dealing with the five fundamental issues raised by mobility:

1. **Locating the mobile host or service:** Before any communication can be initiated, the desired end-point must be located and mapped to an addressable destination.
2. **Preserving communication:** Once a session has been established between end points (typically applications), communication should be robust across changes in the network location of the end points.
3. **Disconnecting gracefully:** Communicating applications should be able to rapidly discern when a disconnection at either end, or a network partition, causes communication to be disrupted.
4. **Hibernating efficiently:** If a communicating host is unavailable for a significant period of time, the system should suspend communications, and appropriately reallocate resources.
5. **Reconnecting quickly:** Communicating peers should detect the resumption of network connectivity in a timely manner. The system should support the resumption of all previously established communication sessions without much extra effort.

[1]An extended position paper on this approach entitled *Reconsidering Internet Mobility* was presented at the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII), Schloss Elmau, Germany, May 2001.

Most current approaches to supporting host mobility provide varying degrees of support for the first two problems. Mobile IP (Perkins, 1996), for example, targets the continuous mobility case, where hosts frequently change attachment point, but always remain connected. The last three components—disconnection, hibernation, and reconnection—have received little attention outside of the file system context (Mummert et al., 1995). We argue that a complete—and useful—solution must address all these issues. For example, today's laptop user may starts her day at home, dialed-in on a modem, plug in to an Ethernet at the office, and perhaps spend time at an airport or coffee shop providing wireless 802.11 connectivity. She should not be forced to restart her network sessions at each stop, nor should the systems she's remotely logged in to be required to expend resources while she's disconnected.

Furthermore, we believe mobility support should be provided at the end hosts. Many previous approaches rely on proxies due to their perceived ease of deployment (Maltz & Bhagwat, 1998; Perkins, 1996). Unfortunately, performance suffers unless the proxies are highly engineered (Maltz & Bhagwat, 1998) and well located in the network, and it becomes difficult to ensure proxies are appropriately located when hosts are mobile (Zenel & Duchamp, 1997). Migrate is an end-to-end approach; no proxies or changes to the IP substrate are required.

We propose to provide mobility services in an optional *session layer* at the end host. This layer presents a simple, unified abstraction to the application to handle mobility: a session. Sessions exist between application-level end points, and can survive changes in the transport, network, and even other session layer protocol states. It also includes basic check-pointing and resumption facilities for periods of disconnection, enabling comprehensive, session-based state management for mobile-aware applications.

An attractive feature of our architecture is that it accomplishes these tasks without sacrificing common-case performance. Migrate provides generic mechanisms for managing disconnections and reconnections in each application session, and for handling application state and context. Un-

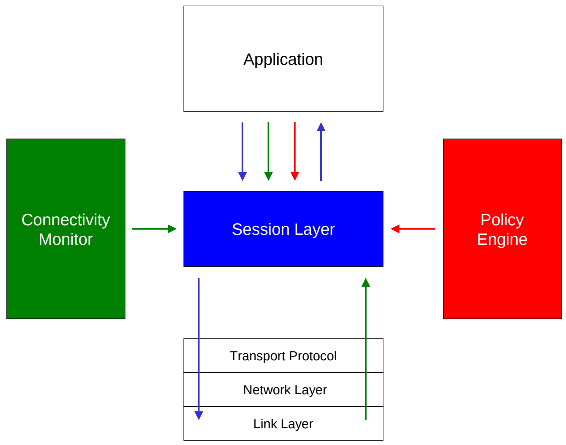

Figure 1. The Migrate session layer framework.

like some previous approaches, however, applications using the Migrate session layer experience minimal overhead except during changes in network attachment point.

2. Architecture

There are many classes of mobile applications: those where other hosts originate connections to a mobile host, those where the mobile host originates all connections, and those where an application-level retry suffices if the network address changes unexpectedly during a short transaction (Cheshire & Baker, 1996). We believe that a good end-to-end architecture for host mobility will support all these modes, and empower applications to make the choice best suited to their needs.

Figure 1 shows the components of the Migrate architecture. There are four interfaces to the session layer: session establishment, connectivity status, policy decisions, and application up-calls. The session establishment API allows applications to specify their notion of a session by explicitly joining together related transport-layer connections (or destinations in connectionless protocols). Once established, a session is identified by a locally-unique token, or Session ID, and serves as the system entity for integrated accounting and management.

Some applications and transport protocols are unable to provide connectivity information at the granularity required for mobility support, hence Migrate supports interfacing with a connectivity monitoring agent which actively monitors the network state between end hosts. The session layer's reaction to disconnection is governed by a system policy engine, enabling the expression of session-specific preferences.

The session layer adapts to disconnection and attachment point changes as needed. The selection of network end point (including the naming scheme used to identify it) and transport protocol, however, remains completely under the application's control. Further, unlike previous network-layer approaches such as Mobile IP (Perkins, 1996), the session layer exposes the specifics of the lower layers to the application through application upcalls, allowing the application to collaborate in policy decisions if it is so inclined.

In a spirit similar to Coda (Mummert et al., 1995), our architecture considers disconnection to be a natural, transient occurrence that should be handled gracefully by end hosts. For extended periods of disconnection, resource allocation becomes an additional concern. While managing application state is outside the scope of our architecture, enabling efficient strategies is decidedly not. In particular, since disconnection often occurs without prior notice, applications may require system support to reclaim resources outside of their control. We are considering a variety of state management services the session layer should implement, including migrating session state between the system and application, and providing contextual validation of session state.

3. Implementation status

We have implemented and evaluated a limited, TCP-specific version of our architecture to support changes in network attachment point accompanied by only brief periods of disconnection (Snoeren & Balakrishnan, 2000). We designed a new end-to-end TCP option to support the secure migration of an established TCP connection across an IP address change. Using this option, a TCP peer can suspend an open connection and reactivate it from another IP address, transparent to an application that expects uninterrupted reliable communication with the peer.

References

Cheshire, S., & Baker, M. (1996). Internet mobility 4x4. *Proc. ACM SIGCOMM* (pp. 318–329). Stanford, CA.

Maltz, D., & Bhagwat, P. (1998). MSOCKS: An architecture for transport layer mobility. *Proc. IEEE Infocom* (pp. 1037–1045). San Francisco, CA.

Mummert, L., Ebling, M., & Satyanarayanan, M. (1995). Exploiting weak connectivity for mobile file access. *Proc. ACM SOSP* (pp. 143–155). Copper Mountain, CO.

Perkins, C. E. (1996). *IP mobility support*RFC 2002).

Snoeren, A. C., & Balakrishnan, H. (2000). An end-to-end approach to host mobility. *Proc. ACM/IEEE Mobicom* (pp. 155–166). Boston, MA.

Zenel, B., & Duchamp, D. (1997). A general purpose proxy filtering mechansism applied to the mobile environment. *Proc. ACM/IEEE Mobicom* (pp. 248–259). Budapest, Hungary.