

CarNet: A Scalable Wireless Network Infrastructure

MIT2000-06

Progress Report: July 1, 2002—December 31, 2002

Robert Morris and M. Frans Kaashoek

Project Overview

The central goal of the CarNet project is to design and build decentralized, self-organizing network systems. Existing networks depend on centralized management at all levels, from deployment of physical infrastructure all the way up to the structure of client/server applications. Centralized architectures are an obstacle in the way of many new uses of networks, including interacting smart devices, peer-to-peer applications, and pervasive deployment in environments without existing infrastructure. The CarNet project addresses this obstacle at two related levels: the Grid network protocol provides communication, and the Chord lookup service organizes distributed applications.

The Grid network protocol automatically organizes radio-equipped nodes into a network. It does this without relying on any pre-deployed infrastructure; in particular, it does not depend on base stations or access points. Instead, Grid nodes cooperatively forward each others' data, with each node essentially acting as a router. This architecture is an ideal way to avoid pre-configuration for smart devices; it also works well when infrastructure is not convenient, as in rooftop radio networks. A major contribution of Grid is its scalable routing protocol, which is built around a peer-to-peer lookup system. Grid also includes power-saving radio control mechanisms for battery-powered nodes, location tracking support for position-aware applications, and hardening techniques to limit the damage caused by malicious or malfunctioning participants. Much of the project effort has gone into the construction and evaluation of a working implementation, consisting of about 30 nodes distributed on multiple floors of the LCS building.

In the last two years we have started a second major area of effort in support for robust peer-to-peer applications. Such applications are appropriate in the same situations as Grid: when centralized servers are not available, or not desirable, and the participants in the system must organize themselves into a cooperative service. The key challenges in this area are scalability, load balance, and robustness in the face of unreliable participants. The current focus of our work is a distributed hash-table algorithm called Chord; this one data structure turns out to be powerful enough to form the basis for a wide variety of peer-to-peer applications. Examples that we are exploring include file sharing, distributed web servers, backup storage systems, DNS-like lookup services, and indexing for fast keyword search.

Progress Through December 2002

In the first half of 2002 we worked on three main areas: investigating robust routing in the face of low-quality wireless links, designing and experimenting with the Ivy read/write peer-to-peer file system, and exploring indexing algorithms for fast keyword search in peer-to-peer file sharing systems.

We finished our analysis of wireless link quality and its impact on existing routing algorithms, and presented a paper describing our results at the HotNets conference. Our core experiment was the following: we ran existing routing algorithms on our wireless test-bed, recorded the throughput of the routes these algorithms chose, found the highest-throughput routes using exhaustive exploration, and demonstrated that the routing algorithms typically choose routes that were a factor of two slower than the best possible route. As a result we have demonstrated

that there is room for improvement, and have produced a benchmark against which we can evaluate the new routing algorithms we're working on.

Our second area of effort was the implementation of the Ivy read/write peer-to-peer file system. Some of our effort involved ensuring that DHash's read/write consistency guarantees were appropriate for Ivy. Ideally DHash would provide simple disk-like behavior in which reads always produced the most recently written data, but that's not possible in a distributed and concurrent system. In the most extreme case, the network might partition with active users in both halves. We changed the design of both Ivy and DHash to make sure that they both detect and sensibly resolve updates in all partitions when the partitions heal. At a higher level, while Ivy attempts to act as much as possible like a file system, it cannot provide exactly the same semantics as a local file disk system, particularly if high performance is desired. Much of our recent effort has involved finding compromises between high performance and strict adherence to ordinary file system semantics.

Our third area of effort has been in efficient keyword search and indexing for peer to peer systems. Our previous peer to peer work (Chord, DHash, CFS, and Ivy) has been oriented towards efficient retrieval of data based on unique keys or names. However, most successful peer to peer file sharing systems (such as Gnutella) provide a keyword search interface to users. Gnutella's implementation of keyword search is arguably inefficient: it broadcasts a copy of every query to every host. We hope to use our efficient key/value storage techniques to implement distributed inverted indices for fast keyword search. The core challenge with this approach is that a search involving multiple keywords (e.g. a search for "robert morris") requires a set intersection between pieces of the index stored on different nodes (e.g. the set of documents containing "robert", and the set of documents containing "morris"). For a naïve design the communication cost would be too large. We have experimented with a variety of techniques to reduce these costs; for example, clustering the documents by topic improves the compressibility of the index sets, even if the clustering is not good enough to (for example) assign sensible topics to documents. We submitted a paper describing our findings to the IPTPS workshop.

Research Plan for the Next Six Months

We plan to continue in all three of the above areas. We will work out the engineering details of a routing algorithm that is aware of wireless link quality, and evaluate it on our test-bed network. We plan to evolve Ivy from a distributed file system into a distributed shared version control system (adopting ideas from CVS), since we suspect that most read/write sharing of files among groups of people is mediated by version control systems. The target users would be participants in geographically dispersed software or document preparation projects. Finally, we will continue to search for better distributed keyword search techniques.

Project Web Sites

<http://www.pdos.lcs.mit.edu/grid/>

<http://www.pdos.lcs.mit.edu/chord/>