EXHIBIT 6

Network Working Group
Request for Comments: 2187
Category: Informational

D. Wessels
K. Claffy
National Laboratory for Applied
Network Research/UCSD
September 1997

Application of Internet Cache Protocol (ICP), version 2

Status of this Memo

Abstract

This document describes the application of ICPv2 (Internet Cache Protocol version 2, RFC2186) to Web caching. ICPv2 is a lightweight message format used for communication among Web caches. Several independent caching implementations now use ICP[3,5], making it important to codify the existing practical uses of ICP for those trying to implement, deploy, and extend its use.

ICP queries and replies refer to the existence of URLs (or objects) in neighbor caches. Caches exchange ICP messages and use the gathered information to select the most appropriate location from which to retrieve an object. A companion document (RFC2186) describes the format and syntax of the protocol itself. In this document we focus on issues of ICP deployment, efficiency, security, and interaction with other aspects of Web traffic behavior.

Table of Contents

1. Introduction

ICP is a lightweight message format used for communicating among Web caches. ICP is used to exchange hints about the existence of URLs in neighbor caches. Caches exchange ICP queries and replies to gather information for use in selecting the most appropriate location from which to retrieve an object.

This document describes the implementation of ICP in software. For a description of the protocol and message format, please refer to the companion document (RFC2186). We avoid making judgments about whether or how ICP should be used in particular Web caching configurations. ICP may be a "net win" in some situations, and a "net loss" in others. We recognize that certain practices described in this document are suboptimal. Some of these exist for historical reasons. Some aspects have been improved in later versions. Since this document only serves to describe current practices, we focus on documenting rather than evaluating. However, we do address known security problems and other shortcomings.

The remainder of this document is written as follows. We first describe Web cache hierarchies, explain motivation for using ICP, and demonstrate how to configure its use in cache hierarchies. We then provide a step-by-step description of an ICP query-response transaction. We then discuss ICP interaction with firewalls, and briefly touch on multicasting ICP. We end with lessons with have learned during the protocol development and deployement thus far, and the canonical security considerations.

ICP was initially developed by Peter Danzig, et. al. at the University of Southern California as a central part of hierarchical caching in the Harvest research project[3].

2. Web Cache Hierarchies

A single Web cache will reduce the amount of traffic generated by the clients behind it. Similarly, a group of Web caches can benefit by sharing another cache in much the same way. Researchers on the Harvest project envisioned that it would be important to connect Web caches hierarchically. In a cache hierarchy (or mesh) one cache establishes peering relationships with its neighbor caches. There are two types of relationship: parent and sibling. A parent cache is essentially one level up in a cache hierarchy. A sibling cache is on the same level. The terms "neighbor" and "peer" are used to refer to either parents or siblings which are a single "cache-hop" away. Figure 1 shows a simple hierarchy configuration.

But what does it mean to be "on the same level" or "one level up?" The general flow of document requests is up the hierarchy. When a cache does not hold a requested object, it may ask via ICP whether any of its neighbor caches has the object. If any of the neighbors does have the requested object (i.e., a "neighbor hit"), then the cache will request it from them. If none of the neighbors has the object (a "neighbor miss"), then the cache must forward the request either to a parent, or directly to the origin server. The essential difference between a parent and sibling is that a "neighbor hit" may be fetched from either one, but a "neighbor miss" may NOT be fetched from a sibling. In other words, in a sibling relationship, a cache can only ask to retrieve objects that the sibling already has cached, whereas the same cache can ask a parent to retrieve any object regardless of whether or not it is cached. A parent cache's role is

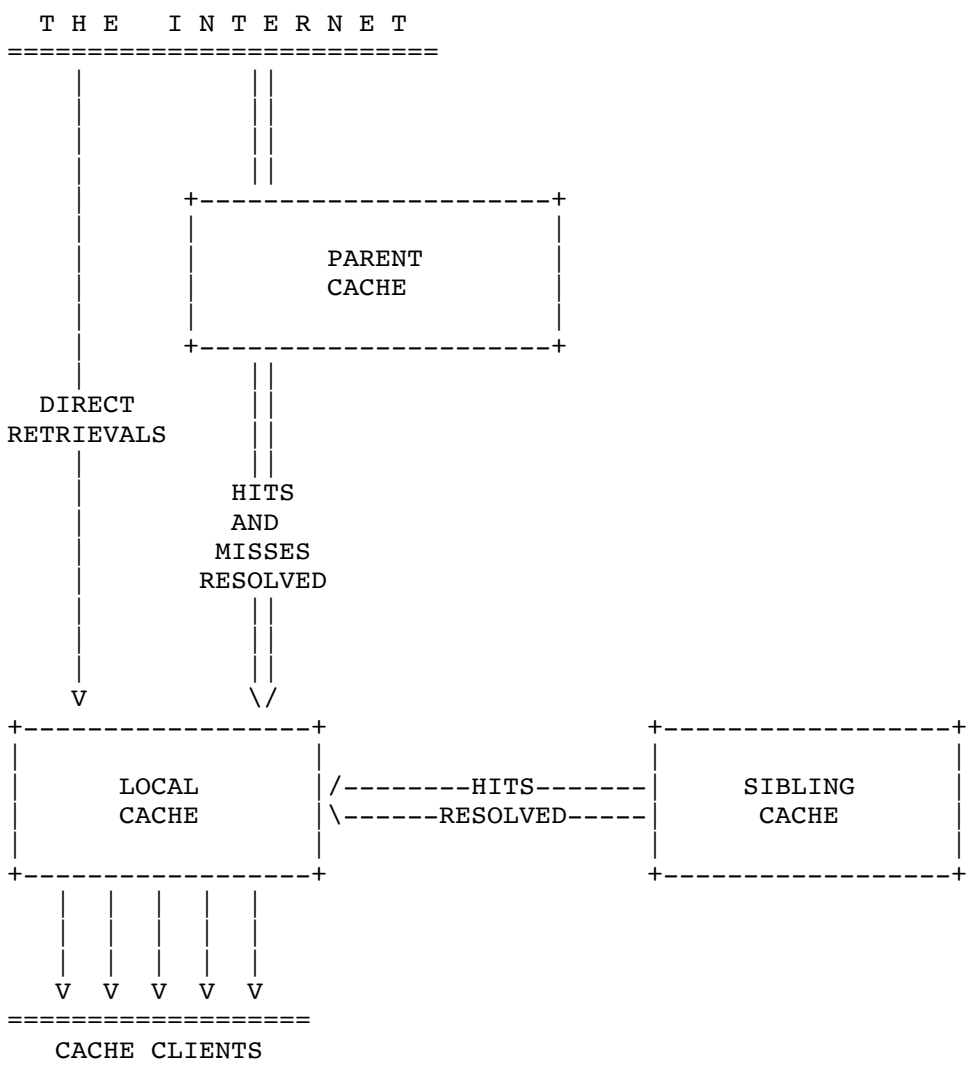

FIGURE 1: A Simple Web cache hierarchy. The local cache can retrieve hits from sibling caches, hits and misses from parent caches, and some requests directly from origin servers.

to provide "transit" for the request if necessary, and accordingly parent caches are ideally located within or on the way to a transit Internet service provider (ISP).

Squid and Harvest allow for complex hierarchical configurations. For example, one could specify that a given neighbor be used for only a certain class of requests, such as URLs from a specific DNS domain.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.