

Network Working Group
Internet-Draft
Expires: August 22, 2001

P. McManus
AppliedTheory Corporation
M. Nottingham
Akamai Technologies, Inc.
February 21, 2001

Requirements for Intermediary Discovery and Description
draft-ietf-webi-idd-reqs-00.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 22, 2001.

Copyright Notice

Copyright (C) The Internet Society (2001). All Rights Reserved.

Abstract

World Wide Web (WWW) intermediaries (such as proxy caches, gateways, and surrogate servers) are widely used by information providers, transport providers, and information consumers to enhance the characteristics of a web transaction. While there are firm models and protocols for the interactions between these devices when they are used, there is no common method used to discover what intermediaries are available. This document establishes a set of requirements for a system that would make discovery of application level intermediaries by web clients efficient and practical.

Table of Contents

| | | |
|---------------------|---|--------------------|
| 1. | Introduction | 3 |
| 2. | Terminology | 4 |
| 3. | Scoping Requirements | 5 |
| 4. | Discovery Requirements | 6 |
| 5. | Description Requirements | 7 |
| 6. | Rationale/Use Cases | 8 |
| 6.1 | Small Network Configuration | 8 |
| 6.2 | ISP Client Configuration | 8 |
| 6.3 | Enterprise Client Configuration | 8 |
| 6.4 | Surrogate Discovery | 9 |
| 6.5 | Automated Mesh Building | 9 |
| | References | 10 |
| | Authors' Addresses | 10 |
| | Full Copyright Statement | 11 |

1. Introduction

Intermediaries are commonly deployed to help scale the WWW. Lack of mechanisms to control and communicate with them brings about scalability issues with intermediaries themselves. Furthermore, access providers who wish to provision caching proxies in their networks have no standardized mechanism to announce such devices to user agents.

As a result of this situation, many network operators resort to the use of interception proxies, which break the end-to-end relationship between client and server at the transport layer, leading to undesired behaviors. Additionally such systems introduce significant cost to override the transport layer and may significantly under-utilize the flexibility of intermediary options that a client has available to it.

This document establishes the functional requirements necessary to build a system for the automatic discovery and description of intermediaries by web clients that they may interact with. It also defines a series of use cases and design goals appropriate to that system.

Requirements for the description and discovery of intermediaries are listed separately in this document, as they comprise two different concepts. It is not assumed that these will, or will not, be separate functions of an eventual system that satisfies these requirements.

2. Terminology

Intermediary - An application level web device that is part of a transaction but is neither the originating or terminating device in the transaction. In HTTP, this includes proxies, surrogates and other gateways but neither User-Agents or Origin Servers.

Discovery Mesh - A logical set of coordinated intermediaries, and their attributes, that reside within a single administrative domain.

Intermediary Discovery - The process of determining and keeping current what devices are in a discovery mesh.

Intermediary Description - The process of conveying to a web client the attributes, roles, and functionality of an intermediary.

IDD Client - A device that seeks to interact with the discovery mesh for either the purposes of providing or obtaining discovery or definition information.

IDD Server - A device that may interact with clients on behalf of a discovery mesh.

3. Scoping Requirements

1. In order to promote modular design and extensibility, some separation in between the discovery and definition aspects of any system fulfilling the requirements of this document should be maintained. This requirement neither prohibits nor requires a single protocol definition.
2. The IDD mechanism should be easy for people with a reasonable amount of knowledge about Web technologies to grasp by leveraging established technologies (e.g. HTTP, XML, URI, etc.) where possible.
3. Server location of "downstream" intermediaries is out of scope.
4. Negotiation of features associated with modification of messages by intermediaries are out of scope.

4. Discovery Requirements

1. The IDD protocol must not require unusual deployment considerations. In particular, IP version 4 broadcast and unicast must form a sufficient operating base.
2. The IDD protocol must provide web clients with a mechanism for locating intermediaries and determining their description properties.
3. The IDD protocol must provide support for an extensible variety of application level web intermediaries. Initially, base support for HTTP, ICP, and RTSP must be provided.
4. The IDD protocol should allow (semi-)automatic configuration in simple networks, with appropriate software.
5. IDD clients must have at least one simple method of interaction that does not require significant state or other processing resources. This method should be suitable for embedded applications.
6. The IDD protocol must be as robust as possible in the event of a network partition of the discovery mesh. This requirement includes the need for well-defined failure modes so that clients have unambiguous information about the state of their request.
7. Significant latency and processing loads to interact with the discovery mesh are only acceptable upon IDD client initialization.
8. The IDD protocol should allow incorporation of externally-derived information where possible (network map, etc..).
9. The IDD discovery mesh should have a mechanism for (semi-)automatic update of itself.
10. The IDD protocol must support a wide diversity of timeliness requirements for IDD clients to learn of changes to the discovery mesh. At a minimum, a range of times from a few seconds to several hours must be supported.
11. The IDD protocol must provide a mechanism for intermediaries to restrict their discovery by IDD clients on an end-to-end basis, without regard to the IDD server.

5. Description Requirements

1. The IDD description mechanism must provide web clients with a method for determining which URIs a particular web client and intermediary pair may resolve.
2. The IDD description mechanism must provide a method for IDD clients to differentiate between surrogates and other more general intermediaries.
3. The IDD description mechanism must provide a method for IDD clients to determine if use of an intermediary is administratively required. If it is required the IDD client must be able to determine a specific path for satisfaction of that requirement.
4. The IDD description mechanism should support determination of "best" intermediary in a multi-faceted extensible way including use of
 - * URI characteristics
 - * Network metrics
 - * Failover options
 - * Intermediary capacity
 - * Intermediary locality
 - * Protocols supported and their specific options (e.g. HTTP method, version, transfer-encodings supported, etc.)
 - * Intermediary authentication and security capabilities
5. The IDD description mechanism should promote efficient use of intermediaries by use of content bucketing, etc..
6. The IDD description mechanism should provide IDD clients with a notion of the freshness and perceived future validity of the data it is serving.

6. Rationale/Use Cases

6.1 Small Network Configuration

A user or access provider wishes to route all HTTP requests through a single proxy. If the proxy is not reachable, requests should be able to be routed directly to their origin servers, but clients should check with the proxy every minute to check its availability.

Deployment should be possible without significant network reconfiguration, preferably by leveraging services already deployed, such as DNS, DHCP, HTTP servers, etc.

The administrative domain in this example is scoped by the /24 network which the proxy serves, which may include both ethernet- and dialup-connected connected clients.

6.2 ISP Client Configuration

An ISP has deployed a cache mesh to control bandwidth requirements. Currently, all requests on port 80 are intercepted and redirected to one of the proxies, but this does not capture all interesting traffic, and causes problems with some applications. The ISP would like to have greater control over proxy load balancing and failover behaviours, and would also like to be able to route certain messages to particular devices, in order to interpose value-added services (such as protocol upgrades, transcoding, ad insertion, etc.).

Deployment must interoperate with the current interception solution with a minimal amount of modification to established services.

The administrative domain in this example is all networks connected to the ISP's border routers, whose scope may be expressed as IP blocks, or as those addresses which reverse-map to a particular domain name.

6.3 Enterprise Client Configuration

An enterprise controls user access to Internet resources by mandating use of proxies at its border firewalls. Additionally, a cache mesh has been deployed internally to reduce internal network load, as well as redistribute internally-sourced content, including both HTTP and streaming.

Deployment should allow clients to locate an appropriate local proxy, and then route messages through the mesh to the appropriate border proxy or internal server, taking into account considerations such as load balancing, failure recovery, client and content locality.

The administrative domain here is all devices inside of the firewall.

6.4 Surrogate Discovery

In any deployment (such as the others listed here), authoritative intermediaries may be present (e.g., surrogates, as part of a CDN). It would be desirable to be able to route appropriate messages to them using IDD, rather than lower-layer mechanisms (such as DNS). This allows tighter integration between the intermediaries and clients, allowing more specific targetting of messages to the appropriate device, without using these mechanisms.

6.5 Automated Mesh Building

Using IDD's description component, an intermediary vendor wishes to specify a system which uses local context (e.g., IP/subnet information) to help generate system configuration, and then combine a number of these configurations to build and manage a caching mesh in a semi-automated manner.

References

- [1] Fielding, R. T., Gettys, J., Mogul, J. C., Nielsen, H. F., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.
- [2] Cooper, I., Melve, I. and G. Tomlinson, "Internet Web Replication and Caching Taxonomy", [RFC 3040](#), January 2001.
- [3] Gauthier, P., Cohen, J., Dunsmuir, M. and C. Perkins, "The Web Proxy Auto-Discovery Protocol", [draft-ietf-wrec-wpad-01.txt](#) (work in progress), July 1999.

Authors' Addresses

Patrick R. McManus
AppliedTheory Corporation
890 Winter Street
Waltham, MA 02451
USA

Phone: +1 781 839-7078
EMail: mcmanus@appliedtheory.com

Mark Nottingham
Akamai Technologies, Inc.
1400 Fashion Island Blvd
Suite 703
San Mateo, CA 94404
USA

Phone: +1 650 627-5247
EMail: mnot@akamai.com

Full Copyright Statement

Copyright (C) The Internet Society (2001). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC editor function is currently provided by the Internet Society.

