Individual Submission
draft-ymbk-opcode-discover-03.txt

Bill Manning
ISI
Paul Vixie
ISC
Erik Guttman
SUN
25 Oct 2001

The DISCOVER opcode

Comments may be submitted to the group mailing list at "mdns@zocalo.net" or the authors.

The capitalized keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119

Abstract:

The QUERY opcode in the DNS is designed for unicast. With the development of multicast capabilities in the DNS, it is desireable to have a more robust opcode for server interactions since a single request may result in replies from multiple responders. So DISCOVER is defined to deal with replies from multiple responders.

As such, this document extend the core DNS specifications to allow clients to have a method for coping with replies from multiple responders. Use of this new opcode may facilitate DNS operations in modern networking topologies. A prototype of the DISCOVER opcode was developed as part of the TBDS project, funded under DARPA grant

F30602-99-1-0523.

Introduction:

This document describes an experimental extension to the DNS to receive multiple responses which is the likely result when using DNS that has enabled multicast queries. This approach was developed as part of the TBDS research project, funded under DARPA grant F30602-99-1-0523. The full processing rules are documented here for possible incorporation in a future revision of the DNS specification."

Method:

DISCOVER works like QUERY except:

1. it can be sent to a broadcast or multicast destination (QUERY isn't defined for non-unicast, and arguably shouldn't be.) While DISCOVER could be used for unicast, what is the point?

2. the Question section, if present, has <QNAME=zonename,QTYPE=SOA> tuples. Future work could augment this structure as follows: <QNAME=service,QTYPE=SRV>

3. if QDCOUNT==0 then only servers willing to do recursion should answer. Other servers must silently discard the DISCOVER request.

4. if QDCOUNT!=0 then only servers who are authoritative for the zones named by some QNAME should answer.

5. responses may echo the request's Question section or leave it blank.

6. responses have "normal" Answer, Authority, and Additional sections. e.g. the response is the same as that to a QUERY. It is desireable that zero content answers not be sent to avoid badly formed or unfulfilled requests. Responses should be sent to the unicast address of the requester and the source address should reflect the unicast address of the responder.

Example usage for gethostby{name,addr}-style requestors:

Compute the zone name of the enclosing in-addr.arpa or ip6.int domain.

DISCOVER whether anyone in-scope is authoritative for this zone.

If so, query these authoritative servers for local in-addr/ip6 names.

If not, DISCOVER whether there are recursive servers available.

If so, query these recursive servers for local in-addr/ip6 names.

So, a node will issue a multicast request with the DISCOVER opcode at some particular multicast scope. Then determine, from the replies, whether there are any DNS servers which are authoritative (or support recursion) for the zone. Replies to DISCOVER requests MUST set the Recursion Available (RA) flag in the DNS message header.

It is important to recognize that a requester must be prepared to receive multiple replies from multiple responders.

Once one learns a host's FQDN by the above means, repeat the process for discovering the closest enclosing authoritative server of such local name.

Cache all NS and A data learned in this process, respecting TTL's.

Usage for SRV requestors:

Do the gethostbyaddr() and gethostbyname() on one's own link-local address, using the above process.

Assume that the closest enclosing zone for which an authority server answers an in-scope DISCOVER packet is "this host's parent domain".

Compute the SRV name as _service._transport.*.parentdomain.

This is a change to the definition as defined in RFC 1034.
A wildcard label ("*") in the QNAME used in a DNS message with opcode DISCOVER SHOULD be evaluated with special rules. The wildcard matches any label for which the DNS server data is authoritative. For example 'x.*.example.com.' would match 'x.y.example.com.' and 'x.yy.example.com.' provided that the server was authoritative for 'example.com.' In this particular case, we suggest the follwing considerations be made:

getservbyname() can be satisfied by issuing a request with this computed SRV name. The servent structure can be populated by values returned from a request as follows:

s_name	The name of the service, "_service" without the preceding underscore.
s_aliases	The names returned in the SRV RRs in replies to the query.
s_port	The port number in the SRV RRs replies to the query. If these port numbers disagree - one of the port numbers is chosen, and only those names which correspond are returned.
s_proto	The transport protocol from named by the "_transport" label, without the preceding underscore.

Send SRV query for this name to discovered local authority servers.

Usage for disconnected networks with no authority servers:

Hosts should run a "stub server" which acts as though its FQDN is a zone name. Computed SOA gives the host's FQDN as MNAME, "." as the ANAME, seconds-since-1Jan2000 as the SERIAL, low constants for EXPIRE and the other timers. Computed NS gives the host's FQDN. Computed glue gives the host's link-local address. Or Hosts may run a "DNS stub server" which acts as though its FQDN is a zone name. The rules governing the behavior of this stub server are given elsewhere [1] [2].

Such stub servers should answer DISCOVER packets for its zone, and will be found by the iterative "discover closest enclosing authority server" by DISCOVER clients, either in the gethostbyname() or SRV cases described above. Note that stub servers only answer with zone names which match QNAME's, not with zone names which are owned by QNAME's.

The only deviation from the DNS[3][4] model is that a host (like, say, a printer offering LPD services) has a DNS server which answers authoritatively for something which hasn't been delegated to it. However, the only way that such DNS servers can be discovered is with a new opcode, DISCOVER, which is explicitly defined to discover undelegated zones for tightly scoped purposes. Therefore this isn't officially a violation of DNS's coherency principles.

IANA Considerations

As a new opcode, the IANA will need to assign a numeric value for the memnonic. The last OPCODE assigned was "5", for UPDATE. Test implementations have used OPCODE "6".

Security Considerations

No new security considerations are known to be introduced with a new opcode, however using multicast for service discovery has the potential for denial of service, primarly from flooding attacks. It may also be possible to enable deliberate misconfiguration of clients simply by running a malicious DNS resolver that claims to be authoritative for things that it is not. One possible way to mitigate this effect is by use of credentials, such as CERT resource records within an RR set. The TBDS project took this approach.

5. Attribution:

This material was generated in discussions on the mdns mailing list hosted by Zocalo in March 2000. Paul Vixie, Stuart Cheshire, Bill Woodcock, Erik Guttman and Bill Manning were active contributors.

6. Author's Address

Bill Manning
PO 12317
Marina del Rey, CA. 90295
+1.310.322.8102
bmanning@karoshi.com

Paul Vixie
Internet Software Consortium
950 Charter Street
Redwood City, CA 94063
+1 650 779 7001
<vixie@isc.org>

Erik Guttman
Sun Microsystems
Eichhòlzelstr. 7
74915 Waibstadt Germany
+49 6227 356 202
erik.guttman@sun.com

7. References

[1] draft-ietf-dnsext-mdns-00.txt
[2] draft-manning-dnsext-mdns-00.txt
[3] RFC 1034
[4] RFC 1035

--bill