EDNS0 deployment

Ronald van der Pol
<Ronald.vanderPol@nlnetlabs.nl>

NLnet Labs
March 2004

Introduction

The deployment of EDNS0 was investigated by analyzing real life traffic from the K and L root nameservers and the .NL ccTLD nameserver. The number of DNS queries to these nameservers with EDNS0 enabled was counted. Data from August/October 2003 was compared with data from February 2004.

Counting the number of EDNS0 enabled queriers

Traces with raw tcpdump data were obtained from the K and L root nameservers and the .NL ccTLD nameserver. These data files were read with tcpdump, selecting packets with UDP port 53 towards the root/ccTLD nameserver. The -vvv verbose output of tcpdump was input for a Perl script (see appendix A). For each packet, this script extracted the source IP address and whether it was an EDNS0 query or not. For each querier, only the first packet was taken into account. Subsequent packets from that querier were discarded (they only contributed to the total number of packets count).

Results

The tables below show the results of counting the number of EDNS0 enabled queriers. The first table are the results from the traces of 2003. The second table are the results from the traces of 2004. The percentage shown is the number of queriers with EDNS0 enabled divided by the total number of queriers. The tables also show how the EDNS0 queries are spread over the various buffer sizes.

There is no data for K @ DENIC for 2003 because that instance of K root did not exist yet at that time.

	K @ AMS-IX	K @ LINX	K @ DENIC	L	.NL
date	20030815	20030815	-	20030825	20031008
EDNS0 queriers edns0/total	29% 6207/21617	30% 7599/24951	- -	21% 9789/46116	24% 27633/116240
EDNS0 bufsize 1280	0.5%	0.6%	-	0.5%	0.5%
EDNS0 bufsize 2048	19%	21%	-	13%	20%
EDNS0 bufsize 4096	9%	9%	-	8%	3%

	K @ AMS-IX	K @ LINX	K @ DENIC	L	.NL
date	20040315	20040315	20040315	20040220	20040224
EDNS0 queriers edns0/total	41% 12645/31015	40% 12196/30812	55% 6012/10968	36% 18182/50398	26% 33425/130739
EDNS0 bufsize 1280	3%	3%	5%	2%	1%
EDNS0 bufsize 2048	29%	28%	37%	25%	22%
EDNS0 bufsize 4096	9%	9%	13%	9%	3%

Additional analysis

Some additional analysis was done. For each trace file the top N of queriers was made. Ignoring the highest queriers did not change the results.

The table below shows what happens when bogus queries are discarded. This was done by (in addition to counting each querier only once) counting only queries with a rightmost label of one of the ccTLDs or a TLD (in addition to queries for '.'). So, queries to foobar.local, etc were discarded.

	K @ AMS-IX	K @ LINX	L
2003 edns0/total	15% 4236/28853	17% 5017/29916	9% 6120/68547
2004 edns0/total	22% 8240/38166	22% 9560/43363	18% 13193/73460

Acknowledgements

The K root trace files were provided by Gerard Leurs, the L root trace files by Steve Conte and the .NL trace files by Jaap Akkerhuis.

Appendix A

```
#!/usr/bin/perl -w

%edns0_hosts = (        # key = IP address, value = # EDNS0 hosts
%tot_hosts = ();        # key = IP address, value = # total hosts
$q_tot = 0;             # total number of queries
$nr_hosts = 0;          # total number of hosts
$nr_edns0_hosts = 0;    # number of EDNS0 hosts
$src = "";              # source IP address of query
$udpsize = 0;           # UDP size in EDNS0 query
%buf = ();              # key = UDP size, value = #queries

$count = 0;


while (<>) {
      $q_tot++;
      # $1: src IP address, $2: EDNS0 bufsize
      if (/^\S+\s+(\d+\.\d+\.\d+\.\d+)\.\d+.*OPT UDPsize=(\d+)/) {
            $src = $1;
            $udpsize = $2;
            # don't count queries from the same host more than once
            next if ($edns0_hosts{$src}++);
            # count the number of various UDP bufsizes
            $buf{$udpsize}++;
      } elsif (/^\S+\s+(\d+\.\d+\.\d+\.\d+)\.\d+/) {
            $src = $1;
      } else {
            print "unknown query: ";
            print;
            next;
      }
      # save all src IP addresses, including those that sent EDNS0 query
      $tot_hosts{$src}++;
}


# total number of unique IP addresses
$nr_hosts = keys %tot_hosts;

# number of unique IP addresses that sent EDNS0 query
$nr_edns0_hosts = keys %edns0_hosts;

print "$q_tot queries from $nr_hosts unique hosts\n";
printf "%d out of %d (%5.2f%%) sent an EDNS0 query\n",
            $nr_edns0_hosts, $nr_hosts, $nr_edns0_hosts / $nr_hosts * 100;
foreach $key (sort {$a <=> $b} (keys %buf)) {
      printf "EDNS0 with UDP size=%d\tqueries=%d\t(%5.2f%%)\n",
            $key, $buf{$key}, $buf{$key} / $nr_hosts * 100;
}

print "\n";
```

```
# print the top N of hosts that sent the most queries
$count = 10;
print "top $count of hosts:\n";
foreach $key (sort {$tot_hosts{$b} <=> $tot_hosts{$a}} keys %tot_hosts) {
        print "$key: $tot_hosts{$key}\n";
        last if ! $count--;
}


# print the top N of hosts that sent the most EDNS0 queries
$count = 10;
print "\n\ntop $count of EDNS0 hosts:\n";
foreach $key (sort {$edns0_hosts{$b} <=> $edns0_hosts{$a}} keys %edns0_hosts) {
        print "$key: $edns0_hosts{$key}\n";
        last if ! $count--;
}
```