

Re: IIS SMTP server in a redundant formation?

Source:

http://www.tech-archive.net/Archive/Internet-Server/microsoft.public.inetserver.iis.smtp_nntp/2009-03/msg00018.htm

- *From:* "Sanford Whiteman" <swhitemanlistens-software@xxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Wed, 11 Mar 2009 03:20:49 -0400
-

In my experience (and many of the others that I talk with) once a client (in this case a sending SMTP server) resolves and successfully connects to a destination SMTP server (MX record) they rarely will connect to a different destination SMTP server (MX record) unless their cached query is pushed out (either expired or flushed).

That is not an accurate real-world generalization of SMTP. The vast majority of SMTP senders do randomize MXs: it is required (a MUST) by RFC (it is not some piddling SHOULD or MAY).

UNreachability is frequently cached, reachability not so. The RFC provides leeway in the first instance (see below).

I /really/ **WISH** that SMTP implementations were as compliant (to the letter or better spirit) of the RFC. Again it is my experience that this is seldom the case. Instead a lot of SMTP implementations will come close but not get as close as you are indicating.

Nope, you have it reversed. Some implementations are broken. Most are not.

Again, if sending SMTP servers were compliant to the spirit of the RFC, yes, however this is not always the case.

It is overwhelmingly the case with this RFC requirement. You must be missing something in whatever you are relying on for evidence here --- maybe one broken MTA vendor?

What some SMTP senders **will** do is cache the IPs of **non-responsive** servers for a hard-coded period, which can delay the rediscovery of downed servers once they have been brought online.

Re: IIS SMTP server in a redundant formation?

Thus going against the spirit of the RFC like I was indicating above.

No, this behavior is seen as allowed by RFC. An SMTP sender may hard-code a future connection order "by recognition of an easily reached address" — usually directly interpreted as "by IP-level route weight or uptime". Caching UNreachability after connection failure has been viewed as reasonable, since it is clearly based on observed downtime.

Excluding an untried MX whether or not you know it's up is a different story: you cannot tell whether an MX is distinctly reachable above other MXs if you don't try other MXs. If all it took to determine easy reachability were a single successful connection, there would be no sense in dictating that senders "MUST randomize [MXs] to spread the load across multiple mail exchangers for a specific organization". Preferring one MX over another of the same weight based solely on a successful previous connection is not allowed. The MTAs that violate this requirement are in the minority.

Based on my memory from reading RFC 821 and it's successor 2821.

Read 5321.

Though the spirit of the RFC strives for what I think you are saying, there have been many many many examples of vendors implementations falling EXTREMELY far short of the RFC.

I wish you would stop saying "spirit" ... we are talking about an IETF MUST requirement.

That's a theoretically valid point and one in favor of NLB... but taking as a given that the appserver uses DNS to locate a smart host (for example, by using a wildcard MX record, as we used to do all the time with sendmail), the setup will do what the poster expects.

Let me re-state: I don't think that your AppServer's .NET or PHP applications will be doing doing MX lookups to decide what server(s) to connect to.

Why not? You can't make that decision for the OP.

Re: IIS SMTP server in a redundant formation?

Rather they will very likely and at most be resolving an IP from a name (A and / or CNAME record), *NOT* be doing MX record processing.

You're cherry-picking to fit your conclusion. The OP already indicated (or believes) that his appservers can or will look up MX records. Otherwise, the discussion of MX records is absolutely off the table — this whole thread is worthless.

Further, if a .NET or PHP application connects to a host name, it is very likely that the server that it is running on is going to use the systems resolver library to resolve the name to an IP(s) and cache the result(s) for future use. (I say "very likely" because I think it unlikely that the average .NET or PHP application is going to do it's own explicit DNS queries, even if it did, it is much less likely to cache the results between invocations.) Thus the .NET or PHP applications are dependent on the systems resolver library to behave correctly.

This is really off-topic, since we are talking about the MX algorithm.

I also call to mind your above statement of "... the client does not further randomize A records under a single MX RR ...". I believe you are referring to the client operating systems's resolver library there.

No, here I am referring only to the SMTP RFC, which specifically prohibits the reordering of A records while specifically prescribing the reordering of MX records.

The prohibition on internally reordering IPs after chasing a single MX has nothing to do with resolver caches. It has to do with SMTP sender implementation alone.

If the resolver library caches the results and does not further randomize A records, you are very likely to get the same A record quite a few times in a row until the cache is either expired or flushed.

The fixed order of A records is irrelevant when it comes to randomizing MXs. That's the idea of not re-randomizing. Whether a set of A records is returned from a resolver cache or passed through from the true DNS server is immaterial to what the SMTP sender does with that list. The sender **MUST NOT** reorder it further.

Re: IIS SMTP server in a redundant formation?

I have found that RFCs are followed when ever it does not impact the design of software and following them will not require (too much) more work than not following them.

We are talking about *one* SMTP RFC requirement: no need to clutter the discussion with grander questions. I know very well about the presumed flexibility of RFCs when it comes to anti-abuse/anti-spam solutions and just about any optional RFC provision. This is not in that larger fuzzy realm. It is a question of basic, real-world MTA operation on an the MXs returned from an MX lookup, based on MUST requirements in an RFC.

--Sandy

.