Picard and Dathon at El-Adrel

Lori MacVittie, 2009-30-04

The importance of context in solving the problems created by tying web applications to deeply rooted local metaphors (IP addresses).

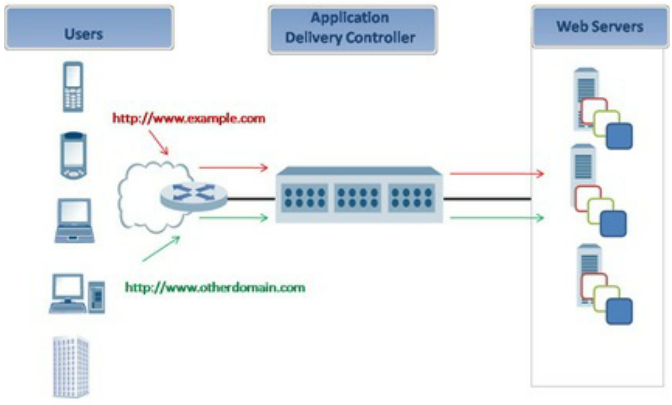
The relationship between IP addresses and web applications to most end-users is much like the metaphorical language of the Tamarians in Star Trek: The Next Generation "Darmok". It is incomprehensible without the proper foundational concepts; to anyone who lacks the proper ***context***. In the case of IP addresses and web applications that foundation is technological rather than the historical basis of the Tamarian's metaphorical language.

The diseconomy of scale inherent in our reliance on IP addresses is well documented. Greg Ness often discusses the growing administrative costs associated with our heavy reliance on the mapping of IP addresses to human-readable host-domain name combinations. But his discussions and illustrations remain primarily focused on the ***internal*** use of IP addresses and reliance on DNS. Rarely is the broader issue of ***external*** reliance on IP addresses discussed or considered.

THE BEAST AT TANAGRA

Roman Stanek recently commented on the issue of IPv4 addresses and cloud computing as it relates to services such as Amazon's EC2. The problem he describes (being assigned an IP address which was previously used by a spammer – or hosting malware – may be blacklisted) could – but is unlikely to – be solved by IPv6. Certainly if Amazon had a wealth of IP addresses available, such as would be the case if we were all living in an IPv6 world, each new customer could be assigned their own individual or range of IP addresses. But eventually a customer is going to move away from the service and the IP address/range of addresses will be reassigned, which could result in the same problem Roman so succinctly describes.

We do not have the luxury of a surplus of IP addresses at the nonce, so the problem of recycling IP addresses remains one with which we must deal. The real issue here is the tight coupling of IP addresses to web applications and users. By assuming a 1:1 relationship in a world where the reality is an M:1 relationship we complicate the issue and make blacklisting and rate-limiting and many other security and performance related functions that rely upon IP address less than optimal. The problem is that we aren't looking at the broader picture, at the ***other*** variables that go into a request – the context – that can help us better determine the nature of a request regardless of the IP address.



There are solutions to the problem, such as taking advantage of *server virtualization (as in network, not host)* and application switching.

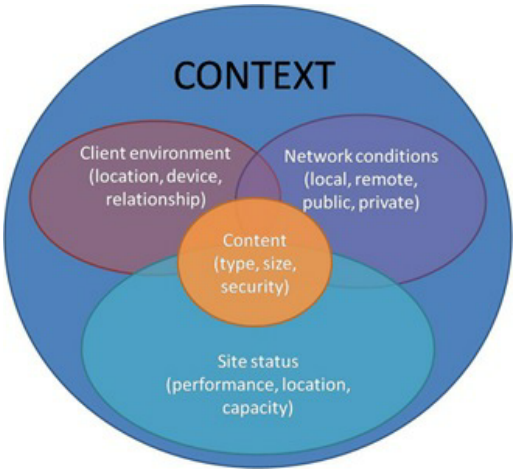
Using application switching (layer 7 load balancing) solves at least the issue of a dearth of IP addresses because all applications are essentially using the same IP address, and the host name and application layer information embedded in HTTP headers and in the application data is used to determine which internal server/service should respond to any given request. This is a fairly straightforward and well understood use of application delivery, one which I take advantage of on a daily basis. Being afforded only a /29 from my local network provider, I need to reuse several IP addresses to host multiple domains. Using layer 7 switching I need only one IP address for all domains and simply use network-side scripting to "switch" which server requests are forwarded to based on the request.

This same principle can be used in cloud computing environments to minimize the number of IP addresses necessary while still affording the ability to host a large number of applications and/or domains.

Unfortunately, this only solves ***one*** problem. It does not address the specific scenario described by Roman in his recent post, nor the broader issues associated with the widespread sharing of a single IP address.

KADIR BENEATH MO MOTEH

The specific problem noted by Roman is that a shared IP address, when abused, can quickly become blacklisted. This essentially makes the IP address unusable and a liability. What's necessary to overcome this problem is the widespread adoption of ***context-aware*** networking. We need to decouple the application from the IP address by expanding the ***context*** of the request and understanding that a dotted quad is about routing to a ***host over a network*** – not necessarily about ***routing a request to an application***. By understanding the context – the host name, the user, etc… – we can mitigate the problems associated with a limited and therefore shared IP address space.



Context-aware networking takes into consideration as many variables as necessary to make decisions about routing, security, and performance. It looks at the user's connection, the user's client information, the application being requested, the state of the infrastructure, cookies – everything it can to better understand how to fulfill a given request based on configured policies. But it also looks at these variables to understand whether the request is legitimate or not, and whether it ***should*** be fulfilled. By examining the context that envelopes a request application delivery looks beyond the IP address as an identifier and can make ***intelligent*** decisions about how to handle each and every request. That's important in a world where a single word – IP address – can represent multiple users or applications.

Using the context of a request makes the old 1:1 relationship between IP addresses and users – and applications – an obsolete method of making security and application delivery-focused decisions.

Captain Picard could not have reached an understanding with Captain Dathon without being able to being to understand the ***context*** of the Tamarian language. The words used by Captain Dathon were understood – much in the same way all infrastructure understands an IP address. But the context surrounding the words were very different when Dathon used them than the way in which Picard used them. To solve their problem they had to share ***context*** and interpret those individual words within a broader set of variables.

Once the crew of the Enterprise understood that they began to be able to effectively communicate with the Tamarians. The same situation is true in application delivery. Once devices understand that they need to evaluate the context in which requests are made, they can better communicate and make the decisions necessary to respond to that request, whether it be to deny based on security policies or respond by routing appropriately to the desired application.

Related articles & blogs:

- Server virtualization versus server virtualization
- Architects need to better leverage virtualization
- Layer 7 Switching + Load Balancing = Layer 7 Load Balancing
- We're sorry. The IPv4 address you are trying to reach has been disconnected
- Can Cloud IP Address Be Damaged Goods?
- The Context-Aware Cloud

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com