

March 2018
Geoff Huston

Stuffing the Camel into the Bikeshed

“Bikeshedding”

Parkinson's Law of Triviality is C. Northcote Parkinson's 1957 argument that members of an organisation give disproportionate weight to trivial issues. He provides the example of a fictional committee whose job was to approve the plans for a nuclear power plant. He postulates that they would spend the majority of their committee time on discussions about relatively minor but easy-to-grasp issues, such as what materials to use for the staff bikeshed, while neglecting the proposed design of the nuclear plant itself, which is far more important and a far more difficult and complex task.

This law of triviality has been applied to software development and other technically focussed activities. The terms “bikeshed effect” and “bike-shedding” were used as metaphors to illuminate the law of triviality; it was popularised in the Berkeley Software Distribution community by the Danish computer developer Poul-Henning Kamp in 1999 and has spread from there to the whole software industry.

https://en.wikipedia.org/wiki/Law_of_triviality

“bikeshedding. The term was coined as a metaphor to illuminate **Parkinson’s Law of Triviality**. Parkinson observed that a committee whose job is to approve plans for a nuclear power plant may spend the majority of its time on relatively unimportant but easy-to-grasp issues, such as what materials to use for the staff bikeshed, while neglecting the design of the power plant itself, which is far more important but also far more difficult to criticize constructively.”

<https://en.wiktionary.org/wiki/bikeshedding>

“Why should I care what color the Bikeshed is? [...] Some people have commented that the amount of noise generated by a change is inversely proportional to the complexity of the change.”

<http://bikeshed.org>

This has become a popular term in the IETF these days, and perhaps deservedly so.

A recent, but by no means unique, example occurred recently in the DNSOP Working group. The proposal was to alter the behaviour of DNSSEC-validating resolvers when processing responses under very particular conditions (<https://tools.ietf.org/html/draft-ietf-dnsop-kskroll-sentinel>) In many ways, the overall intent of the proposal is irrelevant, but the essential item here is the use of a particular string of characters that are used as a trigger label to cause the desired behaviour to occur within the DNS resolver. It appears that more time

and more messages in the working group have been spent on the choice of the particular label to use as the trigger than any other aspect of this proposal. The irony is that the exact choice of characters to use as the label is profoundly irrelevant to the overall design of the proposal. The concept that the actions of every DNSSEC-validating resolver were proposed to be changed in this document appeared to excite very little reaction at all. The issue that the interpretation of the results of any experimentation that used this approach would inevitably be clouded by the long tail of incremental adoption also appeared to excite no reaction at all. A serious debate about the value of any data that could be extracted in this manner against the incremental cost of deploying yet another change in the way DNS resolvers behave was not considered by the Working Group. But the exact label to be used? Well of course, let's talk at length about that. Because it's something everyone knows about, and something that everyone has an opinion on. In such matters of picking the "right" label there is no clear ability to come to a single conclusion, as there is no particular merit in using one label over another. The resultant conversation is a classic example in my view of what "bikeshedding" has come to represent these days.

Many IETF activities encounter the same issue at some time. Often when discussions tend to head towards the choice of names we see the discussion following a familiar path where the assertion is made that the discussion is bikeshedding and the inevitable response that names matter and choice of the right name merits consideration.

"Camelling"

"Possibly from the proverb, recorded by Thomas Fuller in his Gnomologia as "Tis the last feather that breaks the horse's back." This comes from an Arab proverb about loading up a camel beyond its capacity to move, to the scenario of a single straw of grass or hay overwhelming the desert-roaming beast of burden."

https://en.wiktionary.org/wiki/the_straw_that_broke_the_camel%27s_back

We are now seeing another term enter the IETF lexicon of dysfunctional behaviour: "camelling".

Standards bodies exist to create new standards. Leaving things well alone is not part of the proposition here, and the temptation to tweak, poke, prod and massage existing standards is sometimes irresistible.

The DNS is a classic example of this prolonged incremental tweaking. As Bert Hubert pointed out to the DNSOP WG session at the recent IETF 101 meeting when presenting on the topic of "The DNS Camel" (<https://datatracker.ietf.org/meeting/101/materials/slides-101-dnsop-sessa-the-dns-camel-01>), the RFCs that collectively define the entirety of the DNS is now spread out across 185 separate RFC documents. That's 2,781 pages, or 888,233 words! That's either very impressive, or very sad. Or perhaps both at the same time!

The argument Bert makes is that the complexity to implement the complete protocol rises as more features and behaviours are added, and pressure is placed on implementors to add every feature to the code base, irrespective of its level of use or even its potential utility. The DNS patches include the such 'features' as IXFR, DNAME, TSIG, DNSSEC, NSEC3, CDS and CDNSKEY, QNAME minimization, EDNS(0) Client Subnet, DNS Cookies, NSEC caching and DNS over TLS.

Yes, that's an impressive, but bewildering, list of terms and acronyms.

Even by the IETF's own standards the DNS is prolific in the generation of new acronyms and the redefinition of common terms to describe aspects of the DNS. In December 2015 the DNSOP Working Group published RFC

7719, a compendium of all of these terms and their particular and at times peculiar meaning within the context of the DNS. (I am particularly fond of the re-purposing of that very old term “bailiwick” in the definition: “Out-of-bailiwick: The antonym of in-bailiwick.” in RFC7719)

Perhaps even more surprising is that after the publication of RFC7719 a new effort started in January 2016 (yes, within one month of publication!) to update that lexicon of DNS terms (<https://tools.ietf.org/html/draft-ietf-dnsop-terminology-bis-09>).

I was hoping that by referencing this draft that a curious reader could consult this document and gain some further insight as to what this list of terms actually means in the context of the DNS. Alas, even in this impressively long 46-page document, the terms TSIG, CDS and CDNSKEY appear to be missing. The work continues!

This process of incremental refinement of the DNS protocol in the standards arena is not stopping. The IETF is currently contemplating DOH, ANAME, Serve Stale, Catalogue Zones, Multiple Answers, Session Signalling, and Extended Errors, among others. (Yes, see the terminology document if you are wondering what these terms might mean in the context of the DNS.) It could be cynically argued that the closing of the DNS Extensions Working Group in 2013 has stimulated even more proposals to add extensions to the DNS than was the case when there was an IETF Working Group dedicated to just this role.

It is a considerable feat of sustained and intense effort that there are so many good implementations of this protocol out there, and it certainly appears that this outcome is happening despite the apparent efforts of the IETF to confound these implementors.

The folk generating these specifications of extensions may not be doing the DNS any favours. Their motivation in describing behaviours rather than writing code is largely theoretical rather than practical, and they have the freedom to specify how they think the DNS should behave without any grounded reference in how the code base actually behaves, and without any grounded reference in the complexity of the code required to implement each and every new feature of the protocol. These additional features also interact with the existing behaviours and the changes to the DNS demanded by each feature are not simple “bolt on” modules, but often entail rather more delicate and potentially widespread code surgery: As Bert points out, DNAME needs DNSSEC special casing, EDNS Client Subnet impacts cache behaviours, QNAME may trigger additional probing, as does outbound TLS, DNS Cookies, and Multiple answers/qtypes. And speaking personally, I thought I understood NSEC5, but that was only for a few seconds before my brain exploded!

Put simply, Bert Hubert believes that the folk standardizing these DNS features significantly undervalue the consequent incremental cost of implementation and operation of the protocol. He argues that this plethora of new features and new behaviours from the standards effort is unsustainable in the long term. If the DNS code base is the camel, then at some point, we will see the incremental load of the next feature becoming the straw that breaks the camel’s back, or in this case imperils the operational viability of the code base. He argues for a more rational consideration about new features, looking at each of these feature proposals from a cost and benefit perspective. It’s just not sustainable to invent and standardize more and more features and expect that the implementors and operators will willingly underwrite the associated costs each and every time.

Some very thoughtful words were published in February 2018 in RFC8324, by John Klensin. Perhaps it will take some time for these words to permeate their way into the collective consciousness of the DNS standards folk, or, more likely, they will be ignored. Which would be a shame because the document contains useful and timely advice. Anyone who is interested in working in the DNS would find that the time taken to read RFC8324 would find that to be time well spent. Here’s the abstract of that document:

The basic design of the Domain Name System was completed almost 30 years ago. The last half of that period has been characterized by significant changes in requirements and expectations, some of which either require changes to how the DNS is used or can be accommodated only poorly or not at all. This document asks the question of whether it is time to either redesign and replace the DNS to match contemporary requirements and expectations (rather than continuing to try to design and implement incremental patches that are not fully satisfactory) or draw some clear lines about functionality that is not really needed or that should be performed in some other way.

“DNS Privacy, Authorization, Special Uses, Encoding, Characters, Matching, and Root Structure: Time for Another Look?” RFC 8324, John Klensin, February 2018

This disconnection between theory and practice and between cost and benefit lies at the heart of the issue here. It appears that in the DNS, in BGP, in the various incarnations of TLS, and perhaps in many other protocols that the IETF asserts change control, the IETF is not the best protocol steward.

Speaking strictly within the remit of standards, the cost of the specification is just the publication of more text and little else. Little wonder that in so many ways and at so many times, the IETF simply finds it impossible to just say “no” to the torrents of feature requests and additions that incessantly batter at the IETF’s door. Instead, the IETF’s default response is “yes, lets add that to the spec”. In so doing, the IETF is piling ever larger bundles of load onto the back of the protocol camel.

Now we have a word for that behaviour. It’s called “camelling”.

What to do about the Camel in the Bikeshed?

It has to be said that this is by no means a revelation in the standards world. Standards bodies that do not generate new standards are typically seen as moribund and irrelevant, so there is a considerable pressure to keep the output rates high. There is an implicit pressure to keep responding positively to requests to take new concepts and proposals and apply a process of review and revision to create useful and useable technology standards.

However, this environment contains many more actors than just the standards bodies, and the implementors, vendors, service operators, researchers, regulators, and even users all play a role here. The broad set of interests that show up at IETF meetings and on IETF mailing lists are perhaps not only unavoidable, but necessary and valuable in the larger context of producing helpful and relevant standards.

Human nature being what it is, Parkinson’s Law of Triviality is an ever-present issue. Yes, we tend to obsess over trivia and spend way too little time on things that really have consequence and importance. The bikeshedding discussions are just an unavoidable part of what we do, I suspect. While we’re unable to stop it, perhaps we should be a little more aware of our predilection to obsess on the trivial and unimportant and try and prevent such discussions undermining the larger process of careful and considered review of a proposal.

Similarly, the issues with incrementalism and camelling are not going to go away. We are really good at simply adding more onto an existing base. The danger is that we can lose track of the original design parameters and the resultant superstructure is out of balance with the original foundations.

A good example of the dire consequences of such a lack of balance between superstructure and its foundations when making incremental changes can be found in the unfortunate story of the Vasa.

In seventeenth century Sweden, King Gustav II Adolph directed his navy to sign a contract with the shipbuilders of Stockholm to build a warship with two gun-decks, a novel design for Swedish shipbuilders of the time. To achieve this the shipbuilders incrementally “scaled-up” the dimensions of an original single

deck design to meet the length and breadth of this grand new warship, the Vasa. As a consequence of this incremental change, the centre of gravity of the vessel was raised too high for its width, making the vessel unstable.

On the 11th of August 1628, the Vasa left the Skeppsgården shipyard harbor. It had sailed barely two nautical miles when a gust of wind pushed the ship over to its side, pushing open the lower deck gun ports. Water filled the ship and it sank in the harbor, taking 53 lives with it.

I'm sure that there are folk who believe that bodies like the IETF can exercise just the right level of restraint and process management to keep excessive levels of both camelling and bikeshedding out of the IETF and its Working Groups activities.

Speaking personally, I just can't see that happening. At best all we can do is be ready from time to time to look at what we are doing and question why. And when we see critical feedback on the way we work, as in Bert Hubert's presentation to the DNSOP Working Group at IETF 101, or John Klensin's commentary in RFC 8324, we should take the feedback seriously and use it in the constructive manner in which such commentary is intended.

These messages share a common theme: We should ensure that our collective enthusiasm for extending the technology is matched by due care and constraint when proposing changes to the Internet's protocols.

Disclaimer

The above views do not necessarily represent the views or positions of the Asia Pacific Network Information Centre.

Author

Geoff Huston B.Sc., M.Sc., is the Chief Scientist at APNIC, the Regional Internet Registry serving the Asia Pacific region.

www.potaroo.net