A Look Back at A Look Back

Reviewing *Development of the Domain Name System*, 1988 Paper by Paul Mockapetris and Kevin Dunlap

Edward Lewis



IDS 2018 13 July 2018

- An outcome of the discussion over ONION as a reserved domain name that is not a top-level DNS name
 What is the relation of the DNS protocol and domain names?
- What in history led to the current state of affairs?
- ⊙ Are there lessons from the past, overlooked issues still needing to be solved?
- ⊙ It's tempting to make this a history lesson, but the emphasis will be on points made, not the history

Basis of this talk

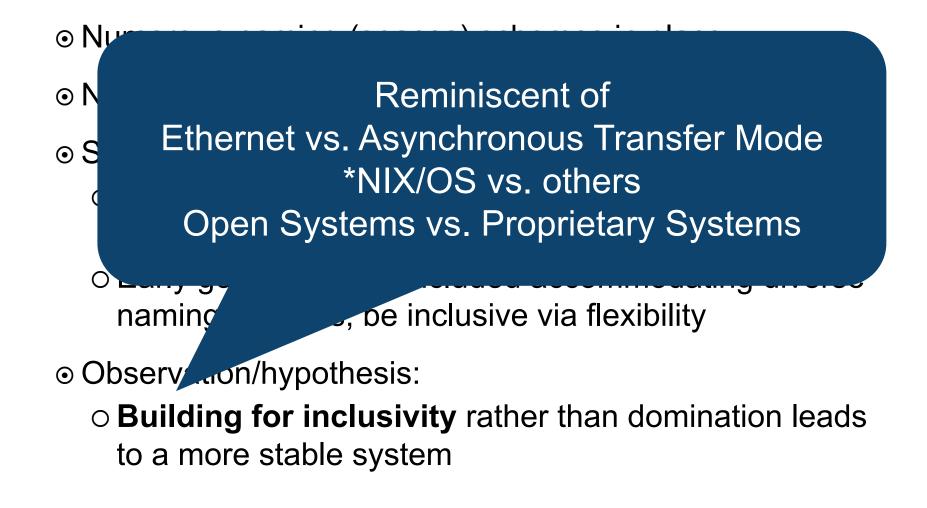
- ⊙ Development of the Domain Name System
 - Originally published in the Proceedings of SIGCOMM '88, Computer Communication Review Vol. 18, No. 4, August 1988, pp. 123–133
- Paul V. Mockapetris USC Information Sciences Institute, Marina del Rey, California
- ⊙ Kevin J. Dunlap Digital Equipment Corp., DECwest Engineering, Washington
- o http://ccr.sigcomm.org/archive/1995/jan95/ccr-9501mockapet.pdf
- Most of the base slide content after slide 5 is copy-&pasted from the paper

⊙ Numerous naming (spaces) schemes in place

- ⊙ Numerous naming (servers) systems in place
- ⊙ Significance
 - There was no name space that the DNS was invented to exclusively implement
 - Early goals for DNS included accommodating diverse naming systems, be inclusive via flexibility
- ⊙ Observation/hypothesis:
 - Building for inclusivity rather than domination leads to a more stable system



From Whence We Came (In the 1980's...)





- ⊙ Be a replacement for HOSTS.TXT
- Maintained in a distributed manner
- ⊙ "Tolerable" performance
- ⊙ Provide extensible services
- ⊙ Avoid trying to force a single style
- **⊙** No obvious size limits
- Interoperate across the DARPA Internet and in as many other environments as possible



Basic Assumptions of the DNS Design

- ⊙ Be a replacement for HOSTS.TX
- Maintained in a distributed mann
- ⊙ "Tolerable" performance
- Provide extensible services
- Avoid trying to force a lingle style
- **No obvious size limits**
- Interoperate across the DARPA Internet and in as many other environments as possible

Over time original limits have been "burned in" to DNS software and into surrounding systems



- ⊙ Be a replace
- \odot Maintained
- ⊙ Provide extension
- ⊙ Avoid trying to for
- The Global Public Internet is not the only DNS, but others seem to be forgotten in standards discussions

- No obvious size limits
- Interoperate across the DARPA Internet and in as many other environments as possible

style



- \odot Size limits ..., limits could be easily changed
- ⊙ Name space structure mirrors the structure of the organization controlling the domain.
- O An administrative decision ... to make the top levels correspond to country codes or broad organization types
- Case-insensitive manner
- Avoid a standard printing rule for names to encourage DNS encoding existing structured names
 - Separated by dots in configuration files, but applications are free to do otherwise
- Decouple structure of the tree from implicit semantics

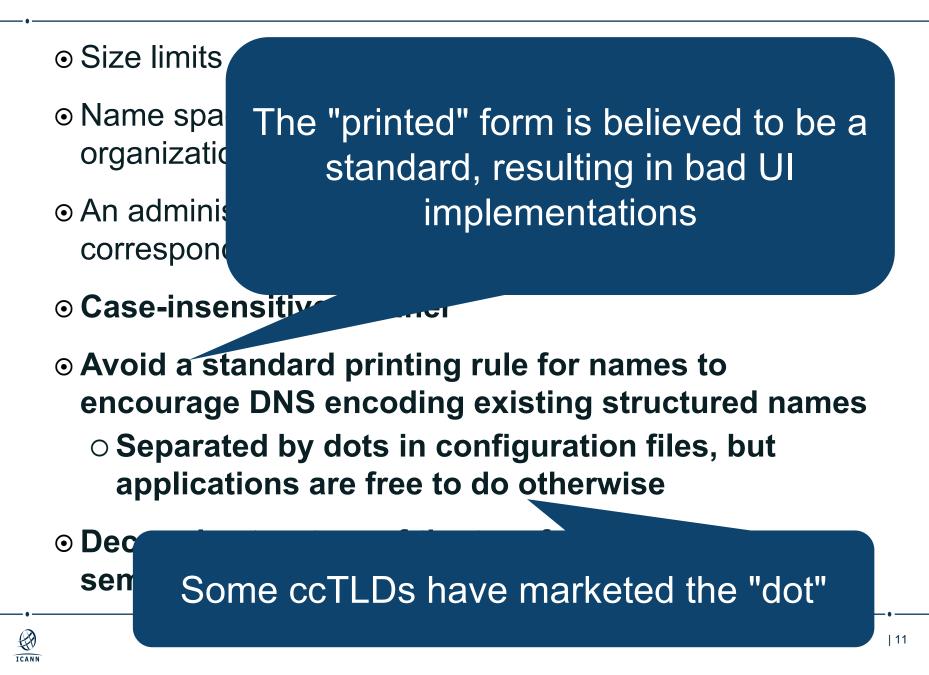
Name Space Assumptions



- Case-insensitive manner
- Avoid a standard printing rule for names to encourage DNS encoding existing structured names
 - Separated by dots in configuration files, but applications are free to do otherwise
- Decouple structure of the tree from implicit semantics



Name Space Assumptions



⊙ Size limits The "underscore" names (started with SRV) are teasing at this ⊙ Name spa organizatio assumption. ⊙ An admini Other times "don't let the protocol correspon shape the tree" referred to • Case-inse assumptions about where data \odot Avoid a st would be stored encourage Dire starea nameS • Separated b m configuration files, but application are free to do otherwise Decouple structure of the tree from implicit semantics

- The class field is meant to divide the database orthogonally from type and specifies the protocol family or instance.
- ⊙ The decision to use multiple RRs of a single type rather than including multiple values in a single RR ... was not a clear choice... suited to use in a limited-size datagram environment
- ⊙ "The recommended TTL value for host names is two days."



 The class field is meant to divide the database orthogonally from type and specifies the protocol family or instance.





- ⊙ Redundant, diverse implementations
- \odot Typical traffic at each on the order of 1 q.p.s
- ⊙ Queries are four types: all information (25 to 40%), host to address (30-40%), address to host (10 to 15%), and new style mail information called MX (less than 10%)
- ⊙ The number of clients is falling as more adopt caching
- **⊙** Static priorities for selecting which root server to use
- Load fluctuations driven by changes in code rather than population
 - \odot 50% of traffic could be eliminated by improvements
 - \odot The root servers refer 10-15% of queries



⊙ Redundant, diverse implementations

- Typical Concerns over selection algorithms
 Querie have been a recurring theme st to address (30-4) stonost (10 to 15%), and new style mail infor called MX (less than 10%)
- ⊙ The number c ments is falling as more adopt caching
- **⊙** Static priorities for selecting which root server to use
- Load fluctuations driven by changes in code rather than population
 - $\odot\,50\%$ of traffic could be eliminated by improvements
 - \odot The root servers refer 10-15% of queries



 Redundant, diverse implementations 	
The idea that code drives load more than user activityhmm. 40%), host to	
address (30-40 %), addre style mail information called	וס נס דס%), and new than 10%)
\odot The number of clients is falling \circ	re adopt caching
 Static priorities for selecting which 	root server to use
 Load fluctuations driven by changes in code rather than population 	
\circ 50% of traffic could be eliminated by improvements	
\odot The root servers refer 10-15% of queries	



98% [initially] from Duane Wessels *Wow, That's a Lot of Packets* (2002) 1 (25 to

i (25 to 40%), host to 0 to 15%), and new

Srmation called MX (less than 10%)

• The er of clients is falling as more adopt caching

- Stat iorities for selecting which root server to use
- Load Juctuations driven by changes in code rather than population

 \odot 50% of traffic could be eliminated by improvements

 \odot The root servers refer 10-15% of queries



styl

- ⊙ Redundant, diverse implementations
- \odot Typical traffic at each on the order of 1 q.p.s
- ⊙ Queries are four types: all information (25 to 40%), host to address (30-40%), address to host (10 to 15%), and new style mail information called MX (less than 10%)
- The number of Referrals at 34% Roy Arends (IDS 2017) including repeated queries
 Load fluctuations drive anges in code rather than population
 50% of traffic could be minated by improvements
 The root servers refer 10-15% of queries



Observations related to Root Servers



Section on Surprises

- It was thought that the semantics of the data was clear, it was not
- Underlying network was much worse than the original design expected, difficulty in making reasonable measurements of DNS performance
- ⊙ The prevalence of "no" answers and the need to cache them
 - Initial monitoring of root server activity showed a very high percentage (20 to 60%) of these responses.
 - The search lists produce a steady stream of bad names



Section on Surprises

⊙ It was thought that the semantics of the data was clear, it was not

 Underlying network was much worse than the original design expected, difficulty in making reasonable measurements of DNS performance

The prevailable
 them
 ○ Initial m
 high pe
 ○ The sea

Noticed when DNSSEC enlarged payload, anycast enlarged capacity and DNS became a "utility" for attackers



Section on Surprises

It was thought that the semantics of the data was clear, it was not
 Underlying design explorements
 Underlying measurements

 The prevalence of "no" answers and the need to cache them

- Initial monitoring of root server activity showed a very high percentage (20 to 60%) of these responses.
- The search lists produce a steady stream of bad names

- Caching but one administrator reversed the TTL and data values, resulting in the distribution of bad data with a TTL of several years ; security of the present system is questionable in an era of local networks and PCs.
- Flexible to accommodate "political" choice; such as to change to the ISO/CCITT directory service
- Datagrams (UDP) much better performance than achieved by TCP
- Variable depth hierarchy ; to encapsulate any system; need to organize
- Additional section to let responder anticipate the next request



Section on Successes

change to

- Caching but one administrator reversed the TTL and data value a TTL of questiona
 Flexible t
- Datagrams (UD better performance than achieved by TC
- Variable depth hierarchy ; to encapsulate any system; need to organize
- Additional section to let responder anticipate the next request



Section on Successes

 Caching – but one administrator reversed the TTL and data values, resulting in the distribution of bad data with a TTL of sev questionab \odot Flexible to Something we seem to have "lost" change to but should look at with DNSSEC... ⊙ Datagram achieve ⊙ Varia′ e de system; need to organize • Additional section – to let responder anticipate the next request

- The type and class data specifiers, which were 8 bits in the draft, should be expanded ; A methodology or guidelines to aid in the design of new types of information is needed
- Needs to be integrated into the operating system to a much greater degree than providing system call to the resolver; specify search lists and defaults in a manner consistent with other system operations
- Demonstrate operational capability before delegating the domain
- Documentation should always be written with the assumption that only the examples are read
- Software versions and parameters should be **accessible**

- ⊙ The type and class data specifiers, which were 8 bits in the draft, should be expanded ; A methodology or guidelines to aid in the design of new types of information is needed
- Needs to be increated into the operating system to a much greater dean.
 reviding system call to the resolver; sr consistent v
- Demonstra domain
- ⊙ Documenta that only th

2004: The TXT vs SPF "incident" Expert Reviews included in 2008 version of *Domain Name System* (DNS) IANA Considerations

• Software versions and parameters should be accessible



)n

ICANN

- The type and class data specifiers, which were 8 bits in the draft, should be expanded ; A methodology or guidelines to aid in the design of new types of information is needed
- Needs to be integrated into the operating system to a much greater degree than providing system call to the resolver; specify search lists and defation a manner consistent with other system operation
- Demonstrate operational capability

ating the

This has happened, and developments like Name Service Switch to integrate DNS with others

But proper search list processing still plagues SSAC Advisory on DNS "Search List" Processing

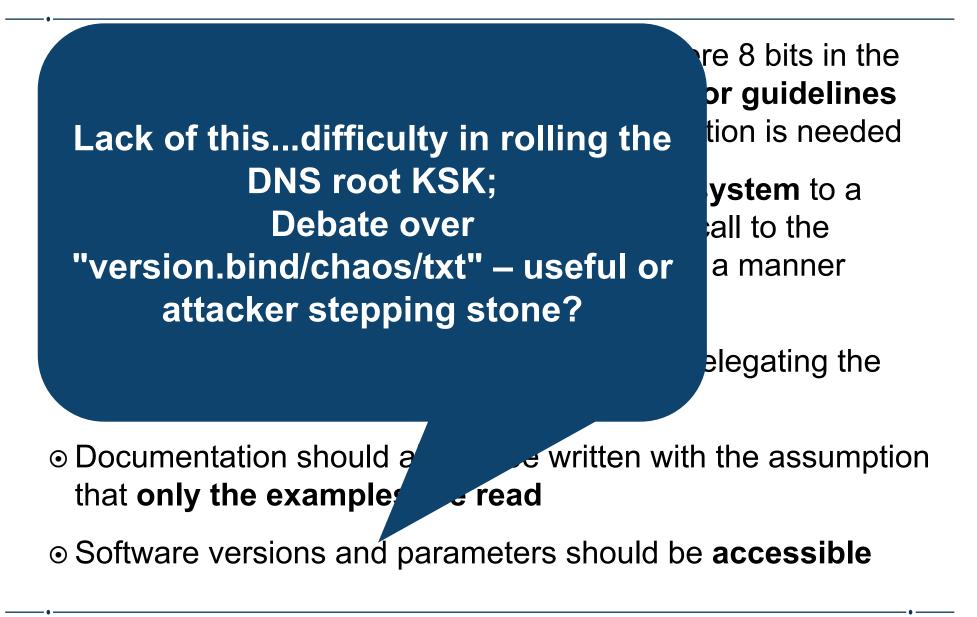
- ⊙ The type an draft, should to aid in the
- Needs to be much greate resolver ; sp consistent with our

Of growing significance, lot of legacy resistance; as new operators are added, less "average experience"

operations

- Demonstrate operational capability before delegating the domain
- Documentation should always be written with the assumption that only the examples are read
- Software versions and parameters should be accessible

- The type and class data specifiers, which were 8 bits in the draft, should be expanded ; A methodology or guidelines to aid in the design of new types of information is needed.
- Needs to be much greate resolver ; sr
 Name Collisions to cite a consequence; problems with some IP address ranges (such as 1.0.0.0/8) consistent with other system operations
- Demonstrate operational capability bef legating the domain
- Documentation should always be written with the assumption that only the examples are read
- Software versions and parameters should be **accessible**



From the Conclusions

- ⊙ Need to distribute functionality was, we believe, inexorable
- ⊙ New functionality and opportunities must be key criteria
- Cache negative responses as well
- ⊙ More difficult to remove functions than get new added
- Variations in the implementation is a great idea; allowing variation in the provided service causes problems.
- Implementors lose interest when system hits initial level
- Distributed software should include a version and table of parameters which can be interrogated
- Systems should include technical means for transferring tuning parameters, or at least defaults, to all installations without requiring the attention of system maintainers

From the Conclusions



- Implementors lose interest when system hits initial level
- Distributed software should include a version and table of parameters which can be interrogated
- Systems should include technical means for transferring tuning parameters, or at least defaults, to all installations without requiring the attention of system maintainers

From the Conclusions



- Implementors lose must t when system hits initial level
- Distributed software should include a version and table of parameters which can be interrogated
- Systems should include technical means for transferring tuning parameters, or at least defaults, to all installations without requiring the attention of system maintainers

⊙ Support for X.500 style addresses for mail, etc

- Tradeoffs between performance, generality, and distribution require at least different styles of use at different levels
- Research in naming systems technical and/or political solutions to the growing complexity of naming will be a growing need.
- Conspicuously absent: Mention of active and/or persistent threats against the stability of the system
 Not a surprise
 - But so completely absent, reflective of "the times"



Engage with ICANN



