ENUM - Experiences

Lawrence Conroy

<lawrence.tel>

Specifications/Acronyms

- **RFC 3401** – DDDS overview
- **RFC 3402** – Core algorithm
- **RFC 3403** – DNS database & Rules (plus some didactic use examples)
- **RFC 3404** – Specification of a general URI generation Service
- **RFC 3405** – URN registration process
- *RFC 2915 – Pre-DDDS NAPTR spec*
- *RFC 2916 – Pre-DNS ENUM spec*
- **RFC 3761** – ENUM specification
- **RFC 5483** – ENUM Experiences
- RFC 3761bis – draft – update to 3761
- Service-guide – draft – definition of process for Enumservice registration

- DDNS = Dyamic Delegation Discovery System
- AUS = (DDDS) Application Unique String
- ERE = (posix) Extended Regular Expression
- URI = Universal Resource Identifiers (rfc3986) ~=~ *URL + URN*
- IRI = Internationalised Resource Identifiers (rfc 3987)
- ADH = Alphabetic, Digit, or Hyphen
- NTN = Non-Terminal NAPTR
- NANP = North American Number Plan

ENUM is…

- A mapping from phone number to different ways of interacting with a user/endpoint

In rfc2916, this was:

- phone-number -> **URI**
  - *This was usually a SIP URI. Not everyone thinks that SIP is the way to connect - SIP providers certainly don't as they block SIP invites from unauthenticated callers so…*

In rfc3761, this is:

- phone-number -> **Enumservice** + **URI**
  - ***Enumservice*** *is the kind of session that results from use of this URI, and the kind of application that's needed for this session*

ENUM is… 2

- ENUM is a DDDS application (3401–3403+/- ½)
- Database/Rule Type: DNS, and NAPTR (3403)
- AUS: Phone number in international format, stripped of all non-digits except initial '+'
- Initial Key Generation/First Well Known Rule:
  - Strip all non-digits
  - Reverse character sequence
  - Intersperse .
  - Append .e164.arpa. domain apex
- Sounds simple – what could possibly go wrong?

ENUM is… 3 - Policy Committees

- Agreed with IAB & ITU-T (a UN organisation)
- ITU-T: each country chooses its own rules
- Name eligibility rules differ
  - Who owns a phone number differs
  - Who can register an ENUM domain differs
  - How they prove eligibility differs
  - What the owner can do with domain differs

"Governance" intrudes

Field Contents: General

- ORDER and PRIO are unsigned integer fields
- Flags, Services & Regexp fields are DNS strings
- Replacement is (uncompressed) FQDN
- Strings interpreted as UTF-8, but…
  - flags & services are case-insensitive printable US-ASCII
  - Regexp ERE shouldn't need non-ASCII (+digits)
    - *Domain owner could put in non-ASCII (e.g. into alternate match strings) but that would be stupid – thus expect someone to do this*
  - Static text in Repl sub-field should be ASCII (URI)
    - *If extended for IRIs, these would need to be pre-processed (URL-escaped)*

Sort ordering

- ORDER is most significant; PRIO is least significant (not the other way around)
- Best value is lowest (not highest)
- Specifications are complex:
  - Are NAPTRs with worse ORDER values ignored?
    - *Clients will use other fields for evaluation*
  - Is PRIO part of the sort or not?
    - *Everything is optional, but clients* do *process it along with ORDER*
  - Do clients consider ORDER/PRIO across domains?

Sort ordering

- Clients have been known to:
  - Ignore ORDER/PRIO entirely, just dealing with the records “as they come”
  - Only take the first record in the RRSet as delivered , ignoring any others
  - Fail, or ignore some or all NAPTRs if they receive a non-terminal NAPTR (usually “lower” ones)
  - Ignore all but NAPTRs with the “winning” ORDER

Flags/Services

- ENUM has one flag: "u" (3404)
  - Service field syntax is:
    E2U 1*(+ ((type) 0*(: (subtype))))
    where type and subtype are each up to 32 ADH
- All DDDS applications inherit "" (3403/4)
  - If flag field is empty, it's a non-terminal rule
- Notes:
  - To avoid collision with 2915/6, E2U is on left – in 2915, it's on right
  - There can be more than one Enumservice; same URI, different actions
  - Service field syntax only works for terminal rules; it's empty in NTNs

Flags/Services

- Clients have been known to:
  - Assume service field is always populated and fail or reject NTNs (sometimes abandoning the query)
  - Fail or misinterpret service fields with more than one Enumservice
  - Reject a NAPTR if they didn't understand or want to use *any* Enumservice in it
  - Process multiple Enumservices in left-to-right or right-to-left order (e.g. voice:tel+sms:tel+fax:tel)
  - Try to scan services field looking for URI schemes

Regexp: ERE and REPL

- DDDS Regexp field (3402 and 3403) generates rule output (*and possibly non-terminal keys*)
- NAPTR Regexp field syntax is:

  <delim>ERE<delim>Repl<delim><iflag>
- Notes:
  - URI output is constructed from Repl field; difficult to find a delimiter character that is **not** valid in URIs, and **not** significant in EREs
  - The ENUM AUS has a character ('+') that is ERE significant so needs to be "escaped" in the ERE sub-field if present
  - The iflag has no effect at all so is pointless to add to ENUM NAPTRs
  - This is UTF-8 in a DNS string; may include any character including **NUL**

Regexp: ERE and REPL

- Clients have been known to:
  - not match the AUS (i.e., expect ERE == ^.*$ or .*)
  - Expect the URI to be static text (i.e., no replacement with sub-expressions)
  - Assume the Delimiter is '!' -- they search the string for '!' as the internal delimiter between the sub-fields
  - Do not deal with REGEXP escaped characters -- they just look for delim character alone
  - Expect the closing delimiter to be the last character in the string
  - ignore everything in NAPTR and look only at the URI

Regexp vs Replacement

- In DDDS, only one is used in any NAPTR:
  - For ENUM output (URI), the only possible field is Regexp
  - For non-terminal output (FQDN), either might be used
    …but…
    In practice, only the Replacement field is feasible to use
- Generalised number-dependent generation of ENUM domains using ERE is impractical/very very hard:
  - Digits in AUS are reversed relative to ENUM FQDN
  - It's easy for NANP numbers, but elsewhere (e.g. in DE, AT, CH) numbers have different lengths, even with common "root" digit patterns
  - Regexp can be a maximum of 255 bytes, so ERE is limited
  - Reasonable maximum length for DNS response limits number of NAPTRs