RPSL implemented in the RIPE database

Cengiz Alaettinoglu <cengiz@ISI.EDU>

David Kessens <davidk@ISI.EDU>

(Slides: ftp://ftp.isi.edu/rps/sanjose-ietf/rpsl-in-RIPE-db.ps.gz)

What is needed ?

- Incompatibilities in representation format in the RIPE database and RPSL
- (necessary) performance improvements
- New objects and attributes
- Dictionary implementation

Incompatibilities in the representation format of the RIPE database and RPSL

- Line continuation
- Comments are allowed everywhere
- Different date format (trivial)
- No reordering of attributes
- Optional syntactic sugar words

Performance improvements

- Goal: close to RPC speed
  - now: pushing the current architecture to the limits without sacrifizing code and data integrity
  - future: redesigned server

Solutions

- Precompute everything we can:

  database file = output format:

  - syntax sugar is included (required for RPSL)
  - only long attributes used
  - regular (-r) queries are handled at maximum speed, -F will stay for compatibility but it's use is discouraged

Smarter indexes

- Index files contain:
  - location
  - size (new)
- Classless index files contain:
  - previously: key for the normal index file
  - now: location and size of the object

Performance comparison

- Tests with all queries of one day (961101, 961102) show an average 2-3 times speed increase
- Not much more is possible without reimplementing the server and/or without breaking correctness

New objects

- route-set
- as-set
- dictionary

New attributes

- member-of
- inject-at
- aggregate-by
- export-components
- members
- members-by-referral
- igp-to-egp
- egp-to-igp
- rp-attribute
- protocol
- typedef
- alias

Syntax checking

- RPSL syntax checking using byacc and perl extensions (Brian Renaud already used this for ‘extended-as-in/out’ attributes)
- definition for hierarchical authorization of set objects
- Correction of common errors:
  - abbreviated IP prefixes to full prefix notation (Example: 10.05/16 -> 10.5.0.0/16)
  - Other convenience translations ?

Dictionary object

- Ordinary object with ‘mnt-by’ attribute
- Initially: object registration only
- Next: syntax checker uses the dictionary object definitions

Conclusion

- RPSL implementation in the RIPE registry software is ongoing
  - performance improvements and general RPSL necessities are ready now
  - syntax checking will be ready before the next IETF
- The RIPE registry software is adequate to start the deployment of RPSL
  - visible changes for the user community will be limited
  - a distributed registry design is underway