RPS Replication

Cengiz Alaettinoglu
cengiz@packetdesign.com
with
C. Villamizar, R. Govindan, D. Meyer, G. Winters

Goals

- Data replication
  - scalable to possibly large number of registries
- Data consistency across registries
  - inconsistent duplicates
  - RPS authorization

Protocol for Replication

- Flooding based
- both pull and push of data
- automatic discovery of registries
  - Registry object
- light−weight option

Replication Mechanisms

- Transaction sequence exchange
  - real time mirroring of transactions
- Transaction flooding
  - unicast
- Repository snapshot
  - complete contents + latest set of transactions

Consistency

- Each repository has
  - refresh
  - expire
- Periodically
  - re−send repository object
    - latest registration sequence numbers
    - time−stamps
- A transaction can not proceed if it depends on an expired repository

Local Registration

- Check
  - object is local
  - RPSL syntax & semantics
  - authentication and authorization
- Hold if
  - a dependent object is missing
  - if an involved registry has expired
- Record version info. (registries, seq#s, ...)
- Accept
- Distribute registration + version−info

Remote Registrations

- Recheck
  - objects local to the remote registry
  - RPSL syntax & semantics
  - authentication and authorization
- Hold if
  - a dependent object is missing or not the right version
- Accept and redistribute
- Check holding registrations