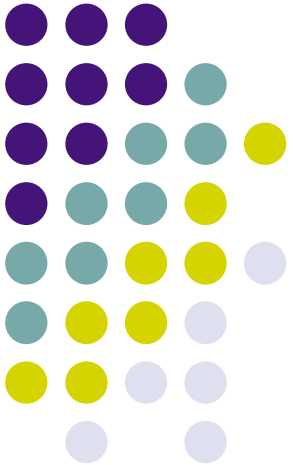
Things to think about when Renumbering an IPv6 network

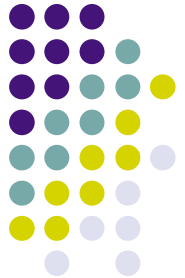
draft-chown-v6ops-renumber-thinkabout-01

Tim Chown, Stig Venaas, Alan Ford, Mark Thompson

{tjc,sv,ajf,mkt}@ecs.soton.ac.uk
University of Southampton (UK)

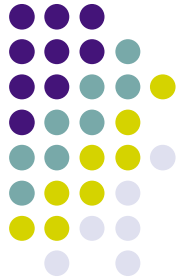


Scope



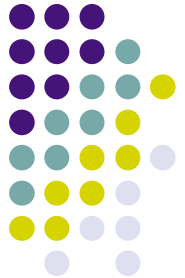
- Authors studying renumbering scenarios, tools and procedures with Cisco/6NET
  - Similar ‘goals’ to old (IPv4) PIER WG c.1996/97
  - Important due to lack of PI space for IPv6 sites
- Complements Baker draft
  - draft-ietf-v6ops-renumbering-procedure-04
  - Defines renumbering procedure and ‘tricks’
  - Experiments to verify Baker procedures for SOHO, enterprise, backbone, ISP
- Our draft focuses on (site) scenarios, requirements, IPv6-specific tools, and issues
  - So identifying other potential areas of work required

Contents (I)



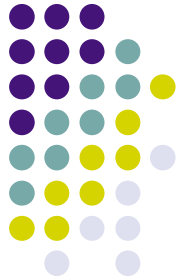
- Scenarios (not all without a flag day)
  - Change of provider (no PI), IPv6 transition (6bone/6to4), topology changes, dynamic prefix, merger/acquisition, mobile networks, cautious initial IPv6 allocations
- Requirements
  - Minimise ‘disruption’: e.g. session survivability, security operation, management tools, DNS data, unreachability
- IPv6 protocol feature considerations
  - Multi-addressing, relationship to multi-homing
  - Mobile IPv6 (e.g. renumbering the HA and MNs)
  - Multicast (e.g. embedded RP, SSM) and (WK) anycast
  - Unique Local Addresses (for internal survivability?)

Contents (II)



- Node configuration issues
  - SLAAC, DHCPv6 with PD, Router Renumbering
- Administrative considerations
  - Router advertisement lifetimes (minimums)
  - Site border filters (firewalls and prefixes, local and remote)
  - Renumbering frequency
  - Delay considerations, freshness of service data, high performance scenarios (e.g. firewalls)
  - Scalability issues - running with 2 (or more) prefixes
  - Dual-stack issues (use of IPv4)
- Impact of topology design
  - Merging networks, fixed length subnets, v6 NAT avoidance

Contents (III)



- Application issues
  - Applications tend not to honour TTLs (API dilemma)
  - Explicit literals in use in apps and system files, or passed as payload data (FTP, etc)
- Summary and ‘call to arms’
  - Need more experimental/procedural results

- Authors feel content is fairly complete
  - Could be distilled if required
- Comments?
  - Mileage in reviving old PIER work?