66nat Engineering Trade-offs

Tony Hain

3/26/09

Engineering Trade-offs

Short term action

1) Define nat66 'because you can'
2) Mangle the IPv6 header
3) Inconsistent mappings
4) Assume only edge deployment
5) Incompatible implementations

Long term impact

1) Increase complexity for app development/deployment
2) Destroys security audit trail
3) Multi-party app referrals will fail
4) Everyone on the path rewrites the header for local optimization
5) Removed over time to avoid cost

Trade-offs - Long term impact vs. short term

- Defining something 'because you can' does not constitute a 'need‘
  - FUD coupled with 'urgency driven by CGN implementations' has led to this knee-jerk reaction bof
  - This entire discussion is simply about 'we can, so we want to no matter what the outcome will be'.
- Working around this proposed impediment will require complex & fragile 3rd-party services to inform the endpoints about how to lie to their peer
  - Maintaining complex traversal mechanisms pushes out implementation of other functions
- Broken/incompatible implementations may actually 'fix' the nat mentality
  - 'barely usable' will entrench them forever because the cost of removing them will be higher than keeping them
  - 'broken' will still see an initial rush, with migration away over time
- If a nat66 work does get chartered, it needs to be experimental
  - nat-pt was forced into stds track by IESG decree

There will be no way to recall this one