The view from IPv6 Operations WG (and we'll talk about translation)

Fred Baker

IPv6 Operations Chair

Author in behave

Ralph and my talks

- Ralph was asked for a view on IPv6 transition/coexistence from the IESG
  - In the absence of someone from softwires, also described the softwires tunneling solutions
- I was asked to speak from IPv6 Operations' perspective
  - Which is to say, to talk about transition/coexistence requirements
  - Oh by the way, I'm working on translation in behave.

V6OPS perspective:

- IPv6 Operations was asked about requirements for transition technologies
  - Other working groups to build the solutions
- Marcelo Bagnulo wrote a requirements document
  - Which the working group rejected as too detailed and ultimately describing a solution instead of giving requirements
  - That said, from the discussion we had a pretty good idea of the requirements and declared the question OBE

First goal: coexistence and transition

1. The point is to get people to turn IPv6 on in their networks
   - While they leave IPv4 on, that is *coexistence*
   - When they turn IPv4 off, that is a *transition*
   - The question is not about IPv4. *It is about IPv6.*
2. Rule of solution suitability
   - If a solution make it desirable for IPv4 to remain on and IPv6 off for an extended period of time, IPv6 has not been turned on.
   - *In this case, see rule 1.*

Second goal: deploy *IPv6*

- Transition technologies fall into two major categories:
  - Those that facilitate IPv6 deployment in a way that when they are no longer necessary we have deployed IPv6
  - Those that change IPv6 “temporarily” in some way, making host changes that will survive the transition
- The latter kinds of technologies do not deploy IPv6
  - They deploy IPv6 with subtle changes that we live with for much longer than we intended

Third Goal: enable communication

- This may seem silly, but it is pretty basic
- Something that doesn’t enable applications to communicate fails to deliver

Fourth goal: reliability, maintainability, servicability

- Operators have to be able to turn it on, diagnose problems, and deliver predictable service to their customers
    - This is both enterprise and service provider

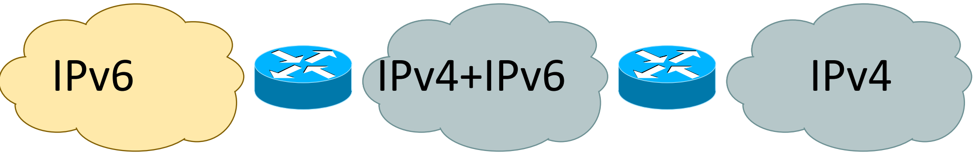
Important lessons from the Internet

- Things we did well
  - The service is **connectivity**
  - Design for **scale** beyond your imagination
  - **Simplicity** is the watchword; elegance and re-usability are keys to both scaling and innovation
  - **Robust Interoperability** is more important than mere correctness
- Things we wish had been done better
  - Avoid design & protocol limitations based on how hardware/technology works today
  - Design for secure channels and secure objects
  - Design for manageability

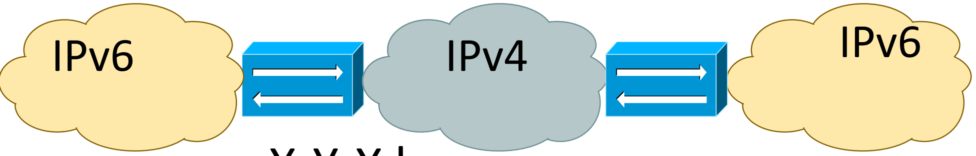
Two broad scenarios

IPv6 applications <-> IPv6
IPv4 applications <-> IPv4

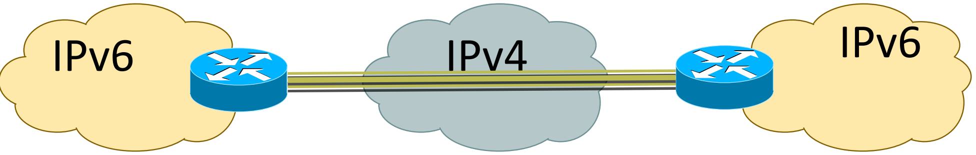
- Three possible approaches
  - Dual stack – ships in the night



  - X-Y-X by translation



  - X-Y-X by encapsulation/tunneling



IPv4 applications <-> IPv6

- Two possible approaches:
  - Stateful translation
    - Similar to IPv4/IPv4 NAT
  - Stateless translation
    - IPv4 address in IPv6 prefix
    - SIIT-like translation
    - NAT-PT deprecated due to scaling issues

X-Y-X scenarios: Comparison to goals

/-X ransla n

Gets IP yed

✔ unchanged

? Enables communication?

? Reliability, Servicability, Maintainability?

- Issue: translation implies

gateway applications for some applications,

Issues similar to IPv4/IPv4 NAT

X-Y-X by encapsulation

✔ Gets IPv6 deployed

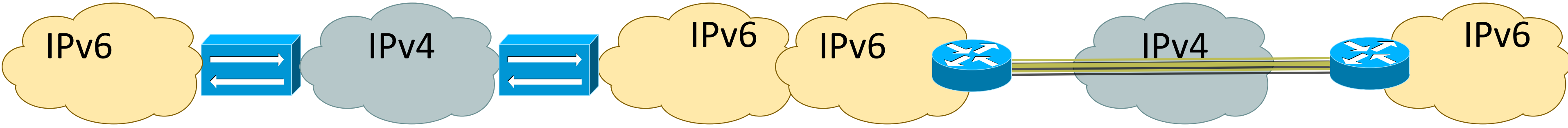
✔ Deploys IPv6 unchanged

✔ Enables communication

✔ Reliability, Servicability, Maintainability

- Issue: standard tunneling/VPN problems in terms of message length

We have solutions for that



Translation scenarios

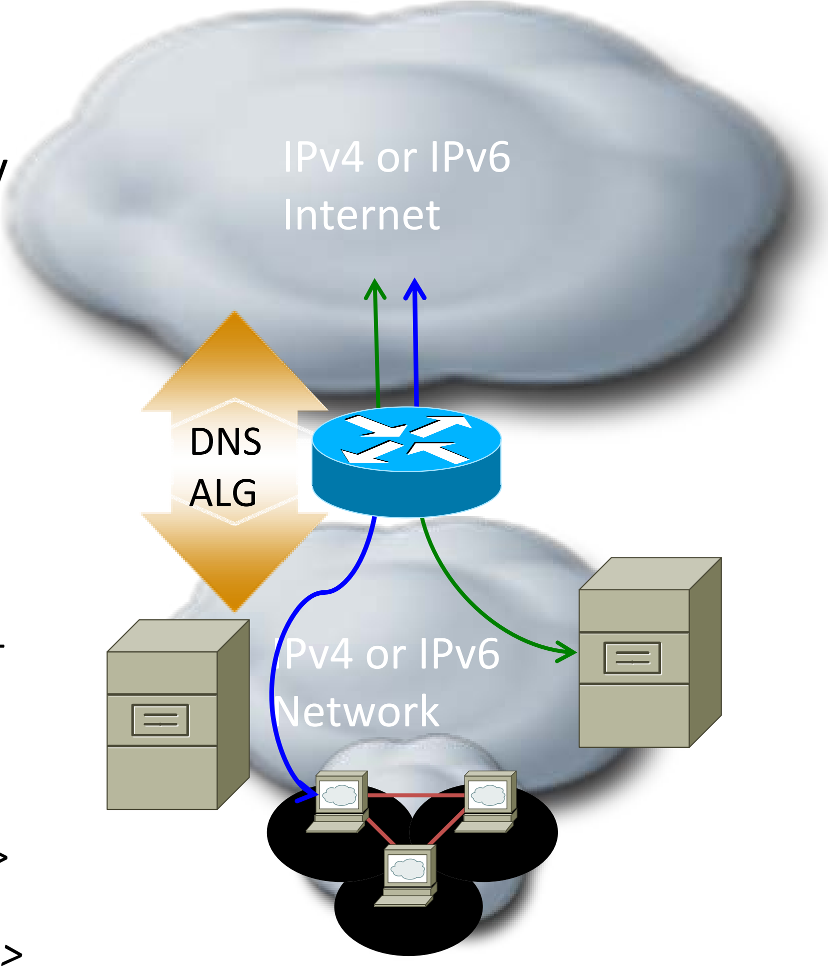
- Objectives:
  - Scalable => stateless if possible
  - Reliable, Maintainable, Servicable => simple to understand and manage
- Would like to be able to initiate sessions:
  - From IPv4-only clients/peers to IPv6-only servers/peers
  - From IPv6-only clients/peers to IPv4-only servers/peers
- Would like to be able to run in edge network and service provider network environments

Issues in existing translation

- NAT-PT:
  - Interaction between DNS and NAT components reduces scalability
- SIIT:
  - Use of a well-known prefix limits routability
  - IPv6 community really likes well-known prefix, but service providers implementing it use a routable prefix
- Traditional IPv4/IPv4 style NAT (NAT64):
  - Ephemeral state in Carrier-grade NAT
  - Initiates sessions IPv6->IPv4 but not IPv4->IPv6

Solution: three components

- DNS64:
  - IPv4 host asks for A records, gets A records
  - IPv6 host asks for AAAA records, may get translated A record
  - No fiddling with NAT tables
- Translator
  - Stateless mode based on CERNET/CERNET2 IVI
    - Modified SIIT algorithm
    - Uses Service Provider Prefix
    - Permits session initiation IPv4 <-> IPv4-mapped-IPv6
  - Stateful mode (NAT64) similar to IPv4/IPv4 NAT
    - Permits session initiation IPv6-native -> IPv4 hosts
    - *Does not permit session initiation IPv4-> IPv6-native*



Oh My Goodness!

What about initiating sessions IPv4-> generic IPv6 address?

Sky falling: whatever shall we do?

First goal: coexistence and transition

1. **The point is to get people to turn IPv6 on in their networks**
   - While they leave IPv4 on, that is coexistence
   - When they turn IPv4 off, that is a transition
   - The question is not about IPv4. It is about IPv6.
2. Rule of solution suitability
   - If a solution make it possible for IPv4 to remain on and IPv6 off for an extended period of time, IPv6 has not been turned on.
   - In this case, **see rule 1.**