ZEROCONF INSIGHTS INTO SCOPING PROBLEMS

Erik Guttman
ZEROCONF WG chair
Sun Microsystems
What is the problem?

• Normally
  – Addresses are relatively stable
  – Names and addresses are unique within the network reachable by a host
  – Datagrams are routable

• We have broken these assumptions with ZEROCONF

• The solutions we have come up with have problems

• Consider
  – IPv4 link-local addresses
  – Link-local name resolution

See: http://www.spybeam.org/issues.html
Ideal Zeroconf Scenario

- Limited number of hosts
- Single link
- Name resolution and/or service discovery provides peer address
- Somewhat more volatile but still pretty stable, unambiguous forwarding, unambiguous names and addresses
Real Zeroconf Scenario

- **L3 issues**
  - Forwarding ambiguity (i3==i2, i2==i1a, etc)
  - Forwarding complexity (i3 is non-LL, i1a is LL)
  - Transitioning (DHCP vs. Zeroconf?)
  - Source address selection

- **L7 issues**
  - Addresses exposed
  - Interface info not used
  - Locators forwarded
  - Renumbering breaks apps
Name Resolution & Discovery Issues

- Scoped locator forwarding
  - Widely done (html &c)
  - resolution may be ambiguous or fail
  - LLMNR: respond per interface
  - RFC 3111: forward locators with scoping in mind (SLP for IPv6)

- IPv6 exposes address scopes via interface indexes – very hard in IPv4
- Existing apps will break in certain scenarios
Solutions and their problems

• **Always maintain a link-local address.**
  Only send LL to LL. But: Legacy interoperation fails, it exacerbates scoping problems and one can‘t turn it off.

• **Transition.** Use global address when possible. But: transition is complicated, leads to instability, forwarding rules become more complex.

• **Round robin resolution.** If at first you don‘t succeed… But: security implications, arbitrary.

• **Higher level ID based forwarding.** Use stable identifier, rediscover peers, control forwarding policy with apps. But: We don‘t know how to do this, no apps do this today.