On “draft-dickson-v6man-new-autoconf”

- Too much in one ID
  - So sorry, it was my first ID
- Will break out bits separately (e.g. allocation bcp)
- To do:
  - Request WG approval on structural (i.e. text) changes to set of RFCs
  - Request WG approval on semantic changes for elements of RFCs
  - Request WG approval on definition/reference changes (break-out a few things)
  - Plus new stuff individually after that, if WG OKs

Structural changes need (IMHO)

- Make set of RFCs easier to read (top down) and understand, esp. to newcomers (there will be many of those in next 3 years)
- Isolate definitions into appropriate/new RFCs
- Structure RFCs so as to enable easier and more consistent implementations
- This is so that not only implementers, but also those using one or more implementations, are better able to build networks (and leads to similar choices of design based on similar criteria for designs)

Minor Semantic Changes

- Remove elements which are no longer relevant
- Take things that are *implicit*, and which currently require full reading of <u>all</u> IPv6 RFCs to fully 'grok', and making them *explicit*, easy to see, stand-alone items
  - e.g. “address constructor” based on Interface Identifier
- Instantiate things that are covered by “blanket” provisions (e.g. Interface Identifier format per media type) into multiple definitions (*currently* redundant, but not necessarily so forever)

Secondary Elements

- The previous items should be suitable in and of themselves to be taken up as WG items
- They enable further enhancements, the ideas of which are in the draft-dickson document
- Discussion of those enhancements may be premature at this point
- Everything I plan on proposing is intended to be 100% backward compatible
- Some RFCs are not flexible in accommodating future work (e.g. new media types)

draft-dickson content stuff

- Mostly as an FYI now...
- To be included in future draft(s)
  - define vanilla “address constructor” as concatenate a /64 prefix with a 64-bit II
  - list current hardware types
    - for each hardware type, list II (currently EUI-64)
  - expand “address constructor” as concatenate /N with M-bit II with leading zero padding, M+N<=128
  - generalize Link Local as FE80::/10 prefix

draft-dickson content (cont.)

- propose modifying II for some media types (e.g. 802.*) to be EUI-48 instead of EUI-64
  - EUI-48 inherently compatible with EUI-64, because OUI reserves FFFE and FFFF values
- The main justification is the bits to the **left** of the prefix/host boundary (subnetting), **not** because of perceived waste on bits to the right
- Allows for much longer doubling *intervals* on exponential growth of the Internet (address consumption, esp. post-IPv4 transition)
- I.e. scaling issues long-term, affect $$$ of ISPs

Bonus Material

- For illustration of low density of OUI values, and to show I'm not trying to squeeze most of the bits out of the host portion of II's:
- We can map 24-bit OUI values into 16-bit values by:
  - 00abcd -> abcd (observe that a is never value 'f')
  - 0abcde (a != 0) -> F000 + (a0c) ^ (bed)
  - abcdef (a != 0) -> F000 + (abc) ^ (efd)
- No duplicates (as of 2007-12-2 oui.txt) !!!

End of Presentation Material

- Slides following this are for anticipated answers to questions from the floor

Impact to RFC 4862

- RFC 4862 – IPv6 stateless autoconfiguration
- With implementation of EUI-48 ii, fully compatible with EUI-64 implementations (which require /64's)
- With other implementations of EUI-48 ii, fully compatible using /3 through /80 prefixes
- Privacy extensions have similar compatibility requirements (backward interoperable on /64's)

Privacy extension thoughts (4941)

- Yes, hides MAC from third parties
- No, doesn't hide existence of use of 4941
- Use of non-OUI, non-EUI-48-in-EUI-64 is very obvious
- Use of EUI-48 does reduce range
- Better to use real OUIs and DAD instead
- Or use DHCPv6 with random requested Ips
- Possibly as an alternative to 4941, with either okay as meeting node requirements

Mix and Match?

- ISPs – assignments to end-users/end-sites
- infrastructure
- hosting
- combo of 4862, 4941, DHCPv6 and static?
- V4 + V6 considerations
  - 1:1 subnets with parallel numbering schemes v4/v6
  - Number of subnets vs size – how many available
  - End-site can use up their v6 space just doing 1:1 !!!

Why Int ID per media type?

- Media *types* are always topologically distinct
- Each type has well-defined hardware address
- Explicitly enumerating allows easier addition of future types
- Canonical approach to doing network drivers
- May better belong in WG(s) for transport
- Mechanism should define behaviour, but not size of type(s) of Int IDs

Prefixes in DFZ – rate issues

- Rate of growth of DFZ is bounded **below** by rate of addition of AS's to DFZ, and by rate of PA/PI blocks assigned to AS's
- PI blocks are likely to be 1 per AS with near-zero growth
- PA blocks are driven by consumption, and only bounded by number of assignments possible from each block
- The smaller the minimum assignment, the longer each block lasts

DFZ rate (cont.)

- The Internet grows on an exponential scale
- The number of end-sites and end-users will eventually saturate
- Before then, growth can expect to continue at exponential rates
- Doubling period is :
  - O(log(PA block size)-log(min block size)
  - i.e. the number of bits between PA and min size
- A goal of 6man/v6ops should be: max this value

Why not a universal II size?

- Does a universal II size/format buy anything?
  - NO
- What uses II?
  - Link Local (which is, by the way, local to the link)
  - 4862 (stateless autoconf) – uses II independent of II format/size, other than how it is currently written (and value-subtract limitations to /64)
  - 4941 depends on 4862 format (only)
  - Nothing else

What does EUI-48 buy?

- Better ability to aggregate hierarchically (ISPs)
- More end-sites per PA block (total)
- Better effects on DFZ, long term
- Applies to 99.9% or more of deployed IP nodes
- Has no direct effect on ability to give/get bigger blocks for end-sites that need them
- ISPs won't go broke in 5 years time
- Makes it easier to get space for 4862/4941!!

It's too late?!?

- If that's the only argument, it's a bad argument
- < 1000 routes including deaggregates in use
- If any of the problems described occur, they will be noticed when it is much worse, e.g. >100,000 prefixes in use
- Better now, with 100% backward compatibility, then later, when we won't **want** backward compatibility (!!)
- Better RFCs will make it easier to build, deploy, and use – needed before 2012 (TEOTWAWKI)

Who are I?

(briand@ca.afilias.info)

- I've been doing IPv4 since 1994, with BGP
- Done >> address allocation stuff, building tools
- Done >> prefix filters (eg, per AS on DFZ peers)
- Know too much about aggregation, allocation, and IGP/EGP issues to ignore scaling problem
- Don't work for an ISP/vendor, no hidden agenda
- Care about long-term health of industry (potential employers)
- Currently do DNS TLDs with IPv6 direct /48's