How shall I address thee?

Let me count the ways...

Karl Auer, IPv6Now

www.ipv6now.com.au

IPv6 Addressing

IPv4 address sources

- Statically configured
- DHCPv4
- Zeroconf (169.254.0.0/16)
- Multiple addresses (on some Oses)
- In general, one scope at a time

IPv6 address sources

- Statically configured
- Autoconfigured
- DHCPv6
- Multiple addresses
- Multiple prefixes
- Multiple scopes
- All at the same time

Dual-stack

- All of the above
- All at the same time
- My **`wlan0`** interface when at home:

```
IPv6 link local :fe80::222:fbff:fe54:1234/64 Scope:Link
IPv6 auto GUA   :2406:a000:0:100:222:12ff:fe54:1234/64
IPv6 auto ULA   :fdae:adad:45d3:1:222:12ff:fe54:1234/64
IPv6 privacy GUA:2406:a000:0:100:d9a8:7251:5b1a:de22/64
IPv6 privacy ULA:fdae:adad:45d3:1:d9a8:7251:5b1a:de22/64
IPv4 RFC1918    :192.168.1.201
```

Autoconf and DHCPv6

- Autoconf *generates* addresses
- DHCPv6 *dispenses* addresses
- IPv6 designed to use BOTH
- Myth: It's one or the other
- But there are still two camps out there

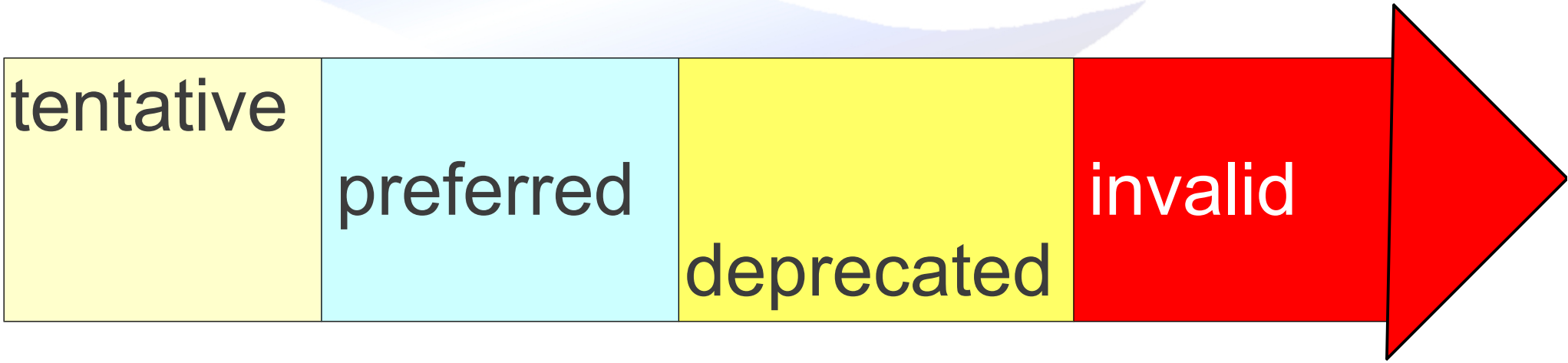
RA *supports* DHCPv6

- RA delivers info for autoconf
- RA can also deliver DHCPv6 hints
- M-flag: “addresses available via DHCPv6”
- O-flag: “info available via DHCPv6”
- Myth: Flags prescriptive, exclusive

Aaaand – it's all dynamic!

- RA-delivered prefixes can come and go
- Autoconf addresses have lifetimes
- DHCPv6 addresses have lifetimes
- DHCPv4 addresses have lease times
- All lifetimes & lease times independent
- Privacy addresses change *all the time*

Lifetimes



preferred lifetime

valid lifetime

Choices, choices

- Which addresses to use?
- Which as the source address?
- Which as the destination address?

- RFC 6724 (was RFC 3484)

Source address selection

- Make a list of all possible candidate source addresses
- Sort the list according to eight rules
- At each comparison, apply rules in order
- If a rule swaps two addresses, stop the comparison
- Use the top address off the list

The application is boss

- If the application specifies a source address, use it
- If anything breaks, the application owns all the pieces
- For example, a DNS server sending responses from a specific address

1: Prefer the same address

- If the destination address is one of your own addresses, use that address as the source address too
- Most operating systems can heavily optimise this situation

2: Prefer appropriate scope

- If you are sending to a link local address, use a link local address as the source
- IPv4 addresses get scope retrofitted!
  - 169.254.0.0/16 → link local
  - 127.0.0.0/8 → link local
  - All others → global

3: Avoid deprecated addresses

- Don't use an address that is about to disappear
- Same principle in other rules, which try to avoid short-lived addresses
- All IPv4 addresses are “preferred” i.e., are never deprecated

4: Prefer home addresses

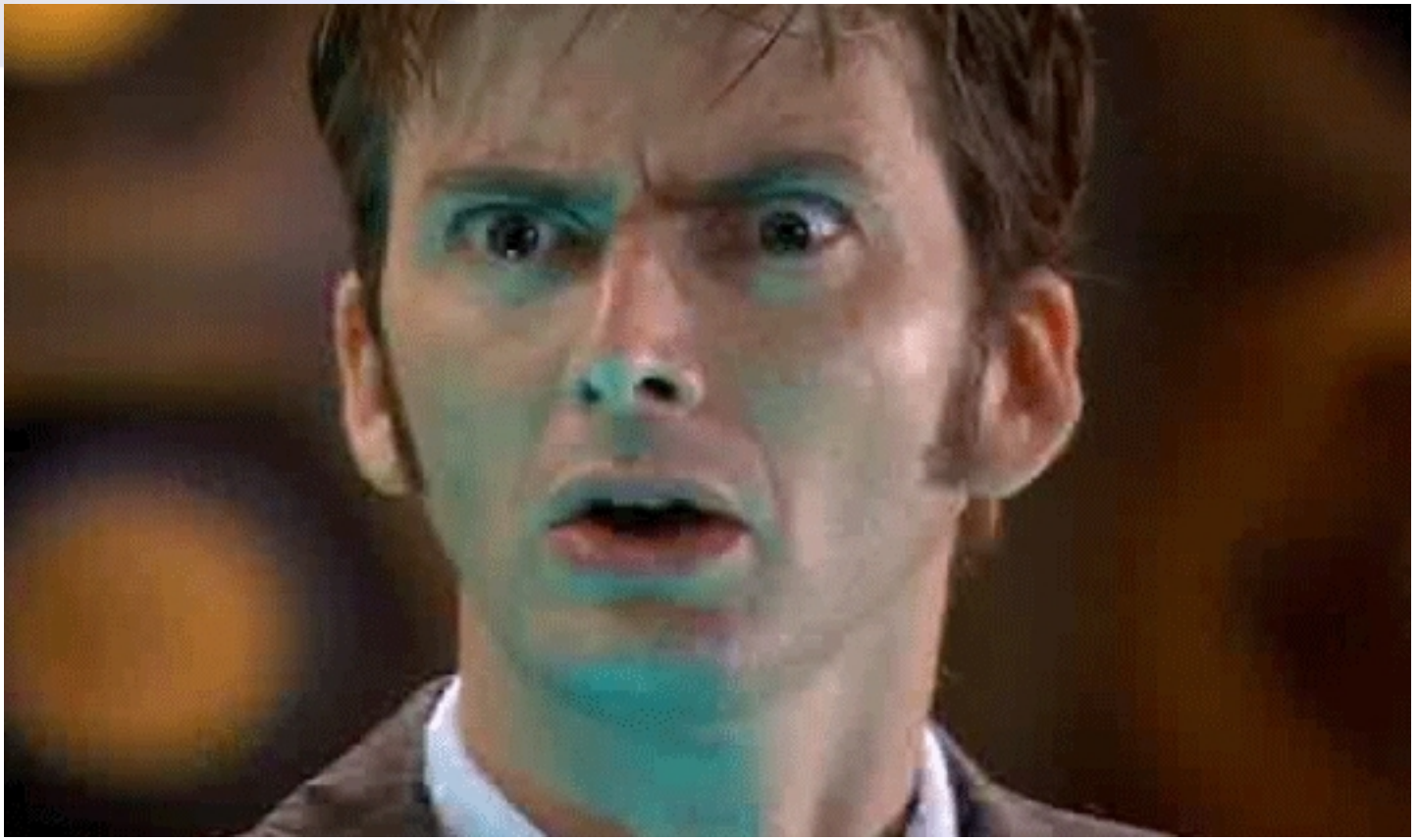
- If you are doing Mobile IP, use your home address rather than whatever care-of-address you happen to have at the moment.

5: Prefer outgoing interface

- The outgoing interface is the interface a packet will be leaving by
- If you have one candidate that is on the outgoing interface and another candidate that is not, prefer the one that is.

6: Prefer matching label

- There is a table that maps certain prefixes to labels
- Look up the destination and the two candidates you are comparing in the table. If one candidate maps to the same label as the destination, and one candidate does not, prefer the one with the matching label.



The label table

Prefix	Label	
::1/128	0	(localhost)
::/0	1	(any IPv6)
::ffff:0:0/96	4	(any IPv4)
2002::/16	2	(6to4)
2001::/32	5	(Teredo)
fc00::/7	13	(ULA)
::/96	3	(IPv4 compatible, deprecated)
fec0::/10	11	(site local, deprecated)
3ffe::/16	12	(6bone, deprecated)

7: Prefer privacy addresses

- If one candidate is a privacy address and the other is not, prefer the privacy address.
- This is new in RFC 6724 – it used to be the other way around, following the principle of preferring longer-lived addresses.
- Implementations will take a while to catch up.

8: Use longest matching prefix

- Prefer the candidate address that shares the most contiguous leftmost bits with the destination
- RFC 6724: Only compare up to the prefix length of the candidate

8: Use longest matching prefix

- Prefer the candidate address that shares the most contiguous leftmost bits with the destination
- RFC 6724: Only compare up to the prefix length of the candidate

IPv6 Now

Destination address selection

- Very similar approach
- Candidates typically come from the DNS
- The entire sorted list is passed up to the application
- The rules are a bit different

1: Avoid unusable destinations

- Rule One is different: “Avoid unusable destinations”
- Up to implementations how to decide whether an address is usable or not
- e.g., no route to it

5: Prefer matching label

- Same as Rule Six in source address selection (and the same table)
- The source addresses being used here (and in other destination selection rules) are those that ***would*** be used ***if*** the candidate destination address were to be used.

6: Prefer highest prededence

- Uses a similar table as for source address selection
- Except it maps prefixes to precedence values instead of labels.

The precedence table

Prefix	Precedence	
::1/128	50	(localhost)
::/0	40	(any IPv6)
::ffff:0:0/96	35	(any IPv4)
2002::/16	30	(6to4)
2001::/32	5	(Teredo)
fc00::/7	3	(ULA)
::/96	1	(IPv4 compatible, deprecated)
fec0::/10	1	(site local, deprecated)
3ffe::/16	1	(6bone, deprecated)

7: Prefer native transport

- As against encapsulated (tunnelled) transports
- Up to the implementation to know which candidates are tied to encapsulating transports
- Only needed for transports with no obvious identifying prefix

8: Prefer smaller scope

- For example, if you have a link local candidate and a global unicast candidate, prefer the link local address

9: Use longest matching prefix

- Compare the leading bits of the candidate addresses and their matching source addresses. Prefer the destination address that share the greater number of identical leading bits.
- RFC 6724: Only compare up to the prefix length of the source address

10: Leave the order unchanged

- This explicitly preserves whatever order two candidates were in at the start, relative to each other, if no other rule has moved them.
- This lets e.g., DNS load balancing work

kauer@ipv6now.com.au
www.ipv6now.com.au

IPv6 Now