An Introduction to IPv6

Owen DeLong
owend@he.net
Acknowledgements

- Special thanks for:
  - Content and graphics:
    - Mukom Akong Tamon (AfriNIC)
    - Nishal Goburdhan (AfriNIC)
  - Research, Data, and graphics
    - Geoff Huston (APNIC)
  - Inviting me to present this
    - Srinivas Chendi (APNIC)
  - Attending
    - All of you
Logistics

- Location of rest rooms
- Please turn off your mobile phone(s)
- Please ask questions
  - Please don’t be afraid to speak up
  - For any non-native English speakers: No matter your English skills, I assure you they are better than my skills at your native language.
History (August 17, 2010)
Why is this important? – Today

IPv4 & IPv6 Statistics
RIR v4 /24s Left
AfrNIC 267,412
APNIC 345,577
ARIN 546,813
LACNIC 263,729
RIPE 332,163
v6 ASNs 8% (3,208/36,955)
v6 Ready TLDs 83% (256/306)
v6 Glues 4,445
v6 Domains 1,457,576

0 days remaining IANA exhausted

Today
RIR Free Pool Projections

Geoff Huston’s math:

Registry Exhaustion Dates

1/31/2010

Hurricane Electric
Why do we need IPv6?
(1. NAT breaks IPSec)

- This works fine until you want to secure the connection with IPSec
- The NAT device modifies the packet header, breaking IPSec
Why do we need IPv6?
(2. NAT Complicates Communications)

- Making this work (through NAT) is a nightmare.
- It’s can be even worse for VOIP
Why do we need IPv6
(3. NAT breaks remote management)

- Branch offices behind NAT are not easily managed from remote locations
Why do we need IPv6
(4. Other IPv4 Limitations)

- Only 3.2 billion global unicast addresses
- Figure at least 5 IP addresses per person
- World Population 6.5 Billion
- Also need addresses for servers, provider infrastructure, etc.
- IPSec implemented as an afterthought mostly at L4
Why do we need IPv6?
(5. Other Implications of NAT)

- Single points of failure -- NAT Gateway keeps state
- Additional hardware resources to maintain state tables/translation tables
- Breaks end-to-end model [p2p, voip, etc.] and hinders incoming connections to inside hosts
- Significant overhead/application bloat dedicated to working around NAT
- Causes problems for audit and abuse identification
- Complicates network troubleshooting and event correlation
Why do we need IPv6? (6 -- The final answer)

- Even with NAT, we’re still rapidly running out of IPv4 addresses.
- IANA ran out in February.
- APNIC may be out as early as June.
- The largest three RIRs (APNIC, ARIN, RIPE NCC) will likely be out* in less than a year, probably before the end of 2011.

* The definition of “out” is subjective and depends on the size block being requested.
Features of IPv6?

- $2^{96}$ times as much address space (3.4x10$^{38}$ addresses total)
- No NAT
- IPSec built into the layer 3 protocol
- Path MTU discovery built in and required to work (don’t indiscriminately block ICMP6)
- Multiple addresses per interface, easier and less disruptive renumbering
- Very large subnets -- no more need to count hosts
Why a new protocol?

- There simply aren’t enough addresses in IPv4
- Restore the end-to-end model of communication enabling significant innovation
  - Smart grids
  - IP Telephony & IPTV
  - Smart devices (TVs, refrigerators, etc.)
  - “Internet of Things”
  - Internet access on planes, trains, and automobiles
- Growing demand for addresses to reach more people
- While we’re at it, fix some (not all) of the issues with IPv4
# A review of the last 14 months in IPv4

## IPv4 Address Space Consumption

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 2010</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
</tr>
<tr>
<td></td>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
</tr>
<tr>
<td></td>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
<td>100</td>
<td>101</td>
<td>102</td>
<td>103</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
<td>108</td>
<td>109</td>
<td>110</td>
<td>111</td>
</tr>
<tr>
<td></td>
<td>112</td>
<td>113</td>
<td>114</td>
<td>115</td>
<td>116</td>
<td>117</td>
<td>118</td>
<td>119</td>
<td>120</td>
<td>121</td>
<td>122</td>
<td>123</td>
<td>124</td>
<td>125</td>
<td>126</td>
<td>127</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>129</td>
<td>130</td>
<td>131</td>
<td>132</td>
<td>133</td>
<td>134</td>
<td>135</td>
<td>136</td>
<td>137</td>
<td>138</td>
<td>139</td>
<td>140</td>
<td>141</td>
<td>142</td>
<td>143</td>
</tr>
<tr>
<td></td>
<td>144</td>
<td>145</td>
<td>146</td>
<td>147</td>
<td>148</td>
<td>149</td>
<td>150</td>
<td>151</td>
<td>152</td>
<td>153</td>
<td>154</td>
<td>155</td>
<td>156</td>
<td>157</td>
<td>158</td>
<td>159</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>161</td>
<td>162</td>
<td>163</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>167</td>
<td>168</td>
<td>169</td>
<td>170</td>
<td>171</td>
<td>172</td>
<td>173</td>
<td>174</td>
<td>175</td>
</tr>
<tr>
<td></td>
<td>176</td>
<td>177</td>
<td>178</td>
<td>179</td>
<td>180</td>
<td>181</td>
<td>182</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>186</td>
<td>187</td>
<td>188</td>
<td>189</td>
<td>190</td>
<td>191</td>
</tr>
<tr>
<td></td>
<td>192</td>
<td>193</td>
<td>194</td>
<td>195</td>
<td>196</td>
<td>197</td>
<td>198</td>
<td>199</td>
<td>200</td>
<td>201</td>
<td>202</td>
<td>203</td>
<td>204</td>
<td>205</td>
<td>206</td>
<td>207</td>
</tr>
<tr>
<td></td>
<td>208</td>
<td>209</td>
<td>210</td>
<td>211</td>
<td>212</td>
<td>213</td>
<td>214</td>
<td>215</td>
<td>216</td>
<td>217</td>
<td>218</td>
<td>219</td>
<td>220</td>
<td>221</td>
<td>222</td>
<td>223</td>
</tr>
<tr>
<td></td>
<td>224</td>
<td>225</td>
<td>226</td>
<td>227</td>
<td>228</td>
<td>229</td>
<td>230</td>
<td>231</td>
<td>232</td>
<td>233</td>
<td>234</td>
<td>235</td>
<td>236</td>
<td>237</td>
<td>238</td>
<td>239</td>
</tr>
<tr>
<td></td>
<td>240</td>
<td>241</td>
<td>242</td>
<td>243</td>
<td>244</td>
<td>245</td>
<td>246</td>
<td>247</td>
<td>248</td>
<td>249</td>
<td>250</td>
<td>251</td>
<td>252</td>
<td>253</td>
<td>254</td>
<td>255</td>
</tr>
</tbody>
</table>

## Allocated to RIR, IANA Free Pool, Other Uses
## A review of the last 14 months in IPv4

<table>
<thead>
<tr>
<th>IPv4 Address Space Consumption</th>
<th>End of 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</td>
<td>Allocated to RIR in 2010</td>
</tr>
<tr>
<td>16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31</td>
<td></td>
</tr>
<tr>
<td>32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47</td>
<td></td>
</tr>
<tr>
<td>48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63</td>
<td></td>
</tr>
<tr>
<td>64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79</td>
<td></td>
</tr>
<tr>
<td>80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95</td>
<td></td>
</tr>
<tr>
<td>96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111</td>
<td></td>
</tr>
<tr>
<td>112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127</td>
<td></td>
</tr>
<tr>
<td>128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143</td>
<td></td>
</tr>
<tr>
<td>144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159</td>
<td></td>
</tr>
<tr>
<td>160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175</td>
<td></td>
</tr>
<tr>
<td>176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191</td>
<td></td>
</tr>
<tr>
<td>192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207</td>
<td></td>
</tr>
<tr>
<td>208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223</td>
<td></td>
</tr>
<tr>
<td>224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239</td>
<td></td>
</tr>
<tr>
<td>240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255</td>
<td></td>
</tr>
</tbody>
</table>

**Allocated to RIR** | **IANA Free Pool** | **Other Uses**
A review of the last 14 months in IPv4

Allocated to RIR in 2010

<table>
<thead>
<tr>
<th>IPv4 Address Space Consumption</th>
<th>End of 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>32</td>
<td>33</td>
</tr>
<tr>
<td>48</td>
<td>49</td>
</tr>
<tr>
<td>64</td>
<td>65</td>
</tr>
<tr>
<td>80</td>
<td>81</td>
</tr>
<tr>
<td>96</td>
<td>97</td>
</tr>
<tr>
<td>112</td>
<td>113</td>
</tr>
<tr>
<td>128</td>
<td>129</td>
</tr>
<tr>
<td>144</td>
<td>145</td>
</tr>
<tr>
<td>160</td>
<td>161</td>
</tr>
<tr>
<td>176</td>
<td>177</td>
</tr>
<tr>
<td>192</td>
<td>193</td>
</tr>
<tr>
<td>208</td>
<td>209</td>
</tr>
<tr>
<td>224</td>
<td>225</td>
</tr>
<tr>
<td>240</td>
<td>241</td>
</tr>
</tbody>
</table>

APNIC receives final normal IANA allocation 31 January, 2011

Allocated to RIR | IANA Free Pool | Other Uses

Monday, October 31, 2011
A review of the last 14 months in IPv4

### IPv4 Address Space Consumption

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
<td>61</td>
<td>62</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>65</td>
<td>66</td>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
<td>79</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>97</td>
<td>98</td>
<td>99</td>
<td>100</td>
<td>101</td>
<td>102</td>
<td>103</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
<td>108</td>
<td>109</td>
<td>110</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>113</td>
<td>114</td>
<td>115</td>
<td>116</td>
<td>117</td>
<td>118</td>
<td>119</td>
<td>120</td>
<td>121</td>
<td>122</td>
<td>123</td>
<td>124</td>
<td>125</td>
<td>126</td>
<td>127</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>129</td>
<td>130</td>
<td>131</td>
<td>132</td>
<td>133</td>
<td>134</td>
<td>135</td>
<td>136</td>
<td>137</td>
<td>138</td>
<td>139</td>
<td>140</td>
<td>141</td>
<td>142</td>
<td>143</td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>145</td>
<td>146</td>
<td>147</td>
<td>148</td>
<td>149</td>
<td>150</td>
<td>151</td>
<td>152</td>
<td>153</td>
<td>154</td>
<td>155</td>
<td>156</td>
<td>157</td>
<td>158</td>
<td>159</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>161</td>
<td>162</td>
<td>163</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>167</td>
<td>168</td>
<td>169</td>
<td>170</td>
<td>171</td>
<td>172</td>
<td>173</td>
<td>174</td>
<td>175</td>
<td></td>
</tr>
<tr>
<td>176</td>
<td>177</td>
<td>178</td>
<td>179</td>
<td>180</td>
<td>181</td>
<td>182</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>186</td>
<td>187</td>
<td>188</td>
<td>189</td>
<td>190</td>
<td>191</td>
<td></td>
</tr>
<tr>
<td>192</td>
<td>193</td>
<td>194</td>
<td>195</td>
<td>196</td>
<td>197</td>
<td>198</td>
<td>199</td>
<td>200</td>
<td>201</td>
<td>202</td>
<td>203</td>
<td>204</td>
<td>205</td>
<td>206</td>
<td>207</td>
<td></td>
</tr>
<tr>
<td>208</td>
<td>209</td>
<td>210</td>
<td>211</td>
<td>212</td>
<td>213</td>
<td>214</td>
<td>215</td>
<td>216</td>
<td>217</td>
<td>218</td>
<td>219</td>
<td>220</td>
<td>221</td>
<td>222</td>
<td>223</td>
<td></td>
</tr>
<tr>
<td>224</td>
<td>225</td>
<td>226</td>
<td>227</td>
<td>228</td>
<td>229</td>
<td>230</td>
<td>231</td>
<td>232</td>
<td>233</td>
<td>234</td>
<td>235</td>
<td>236</td>
<td>237</td>
<td>238</td>
<td>239</td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>241</td>
<td>242</td>
<td>243</td>
<td>244</td>
<td>245</td>
<td>246</td>
<td>247</td>
<td>248</td>
<td>249</td>
<td>250</td>
<td>251</td>
<td>252</td>
<td>253</td>
<td>254</td>
<td>255</td>
<td></td>
</tr>
</tbody>
</table>

---

**Allocated to RIR in 2010**

- January 2010
- February 3, 2011
- 27 February, 2011
- 31 January, 2011

**IANA Free Pool Ends**

- 3 February, 2011

APNIC receives final normal IANA allocation

Monday, October 31, 2011
Applications that require IPv6 (not possible in IPv4)

- Green Tech - Give all energy consuming devices an address and manage/monitor remotely
- Global IP Telephony - No intermediate servers
- Give all users the ability to host services
  - Video streams
  - Personal web site or blog
  - Host games (Console or PC games)
- Natural Disaster warning systems
- Improved access to EMS, Medical telepresence
The choice to be made:

Which Approach will you take?

IPv4/IPv6 Dual Stack Now

My dual stack network is running great!

IPv4 is just fine. We just need more NAT!!
Part 1 – Nuts and Bolts

- First, we’ll teach you about all the little pieces that make IPv6 possible (basics)
- Addresses
- Nomenclature
- Address Structure
- Address Scopes and Purposes
- Etc.
IPv6 -- The basics

1. Address Notation

- 8 groups of 4 hex digits (16 bits) separated by colons (:
- Rules for shortening:
  - Drop any leading zeroes in a group
  - Replace ONE set of consecutive 0 groups with ::
- Example: 2001:0001:0000:0000:00A1:0CC0:01AB:397A

2001:0001:0000:0000:00A1:0CC0:01AB:397A

2001:1:0:0:A1:CC0:1AB:397A

2001:1::A1:CC0:1AB:397A
# IPv6 -- The Basics

## 2. Types of IPv6 Addresses

<table>
<thead>
<tr>
<th>Type</th>
<th>Sub-Types</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unicast</td>
<td>Global Unicast</td>
<td>Currently 2000::/3</td>
</tr>
<tr>
<td></td>
<td>Protocol Use</td>
<td>Uses such as 6to4, Teredo</td>
</tr>
<tr>
<td></td>
<td>Unspecified</td>
<td>Default route or unknown address (::/0)</td>
</tr>
<tr>
<td></td>
<td>Loopback</td>
<td>::1/128</td>
</tr>
<tr>
<td></td>
<td>Link Local</td>
<td>fe80::/16 for use only on local link</td>
</tr>
<tr>
<td>Multicast</td>
<td></td>
<td>Used to implement all ‘broadcast-like’ behavior</td>
</tr>
<tr>
<td>Anycast</td>
<td></td>
<td>Globally Unique address shared by multiple hosts</td>
</tr>
</tbody>
</table>
**IPv6 -- The Basics**

### 3. Address Scopes

<table>
<thead>
<tr>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Global</strong></td>
<td>Routed throughout the entire IPv6 Internet</td>
</tr>
<tr>
<td><strong>Link Local</strong></td>
<td>• Can be used only by nodes on same link</td>
</tr>
<tr>
<td></td>
<td>• Never forwarded by routers</td>
</tr>
<tr>
<td></td>
<td>• Prefix fe80::/10</td>
</tr>
<tr>
<td></td>
<td>• IPv6 hosts automatically generate one per interface</td>
</tr>
<tr>
<td></td>
<td>• Can also be manually configured</td>
</tr>
<tr>
<td></td>
<td>• Requires ZoneID (%interface) for disambiguation</td>
</tr>
<tr>
<td><strong>Unique Local</strong></td>
<td>• Can be used with any site or group of sites on agreement</td>
</tr>
<tr>
<td></td>
<td>• ULA Random (fd00::/8) High probability of uniqueness</td>
</tr>
<tr>
<td></td>
<td>• fc00::/8 Reserved for future use (coordinated ULA?)</td>
</tr>
</tbody>
</table>
IPv6 -- The basics
Anatomy of a Global Unicast address

<table>
<thead>
<tr>
<th>3 bits</th>
<th>9 bits</th>
<th>20 bits</th>
<th>16 bits</th>
<th>16 bits</th>
<th>64 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>IANA to RIR</td>
<td>RIR to ISP</td>
<td>ISP to End Site</td>
<td>Net</td>
<td>Interface ID</td>
</tr>
<tr>
<td>001</td>
<td>IANA to RIR</td>
<td>RIR to End Site</td>
<td>Net</td>
<td>Interface ID</td>
<td></td>
</tr>
<tr>
<td>3 bits</td>
<td>9 bits</td>
<td>36 bits</td>
<td>16 bits</td>
<td>64 bits</td>
<td></td>
</tr>
</tbody>
</table>

- Every end site gets a /48
- Global Unicast currently being allocated from 2000::/3
  - Top: Provider assigned
  - Bottom: Provider Independent
IPv6 -- The basics
Anatomy of a Link Local address

- Always fe80::/64, re-used on every link
- Never forwarded off-link (link scope)
- Must be present for interface to participate in IPv6, Automatically configured
- ZoneID used to disambiguate different links
Recall: every link has link local address in \texttt{fe80::/64}

So out which interface should the router send out a packet to \texttt{fe80/64}?

ZoneIDs (or scopeIDs) disambiguate by specifying the interface to which the address belongs.

Specified as [RFC4007]: \texttt{address\%zoneID}

ZoneID could be the interface number or other identifier.

\texttt{e.g ping fe80::245:bcff:fe47:1530\%fxp0} [on a FreeBSD system]
IPv6 -- The basics
Anatomy of Unique Local Addresses [RFC4193]

- Like RFC1918 but lower chance of collision
- Prefix fc00::/7 + “L flag” (bit 8) indicates whether prefix is locally assigned [1] or globally assigned [0]. (Global currently deprecated)
- Scope is global, but, filtered by most routers and ignored by most eBGP peers
- fd00::/8 Global IDs generated at random (RFC4198)
- Uniqueness is NOT guaranteed but highly likely
IPv6 -- The basics
Anatomy of a 6to4 Transition Address

<table>
<thead>
<tr>
<th>48 bits</th>
<th>16 bits</th>
<th>64 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002</td>
<td>WWXX:YYZZ IPv4 Address</td>
<td>SubnetID</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Interface ID</td>
</tr>
</tbody>
</table>

- WWXX:YYZZ is the hex form of a public IPv4 address
- Used to create global prefixes and hosts for v6 capable sites on the IPv4 internet
- Better in theory than practice
- Example address: 192.159.10.200/32 (IPv4) becomes 2002:c09f:0ac8::/48 (65,536 IPv6 subnets for every IPv4 host)
IPv6 -- The basics
Anatomy of an IPv4–Mapped Transition Address

- Usually represented as ::ffff:w.x.y.z where w.x.y.z is a normal address
- Internally represents an IPv4 node to an IPv6 node
- Never used on the wire
- Simplifies software development for dual-stack (treat all connections as IPv6)
### IPv6 -- The basics

#### ISATAP Transition Addresses

<table>
<thead>
<tr>
<th>64 bits</th>
<th>32 bits</th>
<th>32 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix</td>
<td>0000:5efe</td>
<td>Private IPv4 Address</td>
</tr>
</tbody>
</table>

- 64 bit prefix (fe80:: for link local, others depend on ISATAP server being used)
- Patented by SRI, License free
- Most widespread implementation of ISATAP is Teredo
- Brittle and hard to troubleshoot
- A poor substitute for native IPv6 connectivity or better tunnel choices
IPv6 -- The basics

Anatomy of a Multicast Address

<table>
<thead>
<tr>
<th>8 bits</th>
<th>4 bits</th>
<th>4 bits</th>
<th>112 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>ff</td>
<td>Flags</td>
<td>Scope</td>
<td>GroupID</td>
</tr>
</tbody>
</table>

- Flags determine whether the address is transient, based on a unicast prefix, or embeds an RP address
- ff00::/8
- Permanent groupIDs are independent of scope while transient groupIDs are only relevant within their scope.
## Flag and Scope Bits in Multicast Addresses

### The Flag Field

<table>
<thead>
<tr>
<th>Bit Position</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 3</td>
<td>Undefined</td>
</tr>
<tr>
<td>Bit 2 (R flag)</td>
<td>Rendezvous Point address is embedded (1) or not (0)</td>
</tr>
<tr>
<td>Bit 1 (P flag)</td>
<td>Address is based on a unicast prefix (1) or not (0)</td>
</tr>
<tr>
<td>Bit 0 (T flag)</td>
<td>Address is IANA defined (0) or transient (1)</td>
</tr>
</tbody>
</table>

### The Scope Field

<table>
<thead>
<tr>
<th>Binary</th>
<th>Hex</th>
<th>Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>1</td>
<td>Interface</td>
</tr>
<tr>
<td>0010</td>
<td>2</td>
<td>Link</td>
</tr>
<tr>
<td>0100</td>
<td>4</td>
<td>Admin</td>
</tr>
<tr>
<td>0101</td>
<td>5</td>
<td>Site</td>
</tr>
<tr>
<td>1000</td>
<td>8</td>
<td>Organisation</td>
</tr>
<tr>
<td>1110</td>
<td>E</td>
<td>Global</td>
</tr>
<tr>
<td>Others</td>
<td></td>
<td>Unassigned or Reserved</td>
</tr>
</tbody>
</table>
# Some Reserved/Well Known Multicast Groups

<table>
<thead>
<tr>
<th>Address</th>
<th>Scope</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FF01::1</td>
<td>1=Interface</td>
<td>All nodes on the interface</td>
</tr>
<tr>
<td>FF02::1</td>
<td>2=Link</td>
<td>All nodes on the link</td>
</tr>
<tr>
<td>FF01::2</td>
<td>1=Interface</td>
<td>All routers on the interface</td>
</tr>
<tr>
<td>FF02::2</td>
<td>2=Link</td>
<td>All routers on the link</td>
</tr>
<tr>
<td>FF05::2</td>
<td>5=site</td>
<td>All routers in the site</td>
</tr>
<tr>
<td>FF02::5</td>
<td>2=Link</td>
<td>All OSPFv3 routers</td>
</tr>
<tr>
<td>FF02::6</td>
<td>2=Link</td>
<td>OSPFv3 designated routers</td>
</tr>
<tr>
<td>FF02::A</td>
<td>2=Link</td>
<td>All EIGRPv6 routers</td>
</tr>
<tr>
<td>FF02::D</td>
<td>2=Link</td>
<td>All PIM routers</td>
</tr>
<tr>
<td>FF02::1:FFXX:XXXX</td>
<td>2=Link</td>
<td>Solicited-node address</td>
</tr>
</tbody>
</table>
Solicited-Node Multicast Address

- Used as destination for Neighbor Solicitation
- Constructed from prefix ff02::1:ff00:0/104 and the last 24 bits of the IPv6 address
- Every node will listen to and respond to its solicited node address
- Allows link-layer address resolution to not disturb (most) unconcerned nodes.

![Diagram of Solicited-Node Multicast Address]
## IPv6 – The Basics

### Address Type Rehash

#### Summary of IPv6 Address Types

<table>
<thead>
<tr>
<th>TYPE</th>
<th>STRUCTURE</th>
<th>TYPE</th>
<th>STRUCTURE</th>
<th>TYPE</th>
<th>STRUCTURE</th>
<th>TYPE</th>
<th>STRUCTURE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>16 BITS</td>
<td></td>
<td>16 BITS</td>
<td></td>
<td>16 BITS</td>
<td></td>
<td>16 BITS</td>
</tr>
<tr>
<td>Global Unicast</td>
<td>Global ID</td>
<td>Subnet ID</td>
<td>Interface ID</td>
<td>ULA-Reserved</td>
<td>fcxx</td>
<td>0</td>
<td>Subnet ID</td>
</tr>
<tr>
<td>ULA-Reserved</td>
<td></td>
<td></td>
<td></td>
<td>fxxx</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ULA-Local</td>
<td>fxxx</td>
<td>0</td>
<td>Subnet ID</td>
<td>Interface ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Link-Local</td>
<td>fe80</td>
<td>0</td>
<td></td>
<td>Interface ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IPv4 Mapped</td>
<td>0</td>
<td></td>
<td>ffff</td>
<td>&lt;IPv4 Address&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6to4</td>
<td>2002</td>
<td>&lt;IPv4 Address&gt;</td>
<td>Subnet ID</td>
<td>Interface ID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISATAP</td>
<td>&lt;64bit v6 Prefix&gt;</td>
<td>0</td>
<td>5efe</td>
<td>&lt;IPv4 Address&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Unspecified</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Loopback</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multicast</td>
<td>ff&lt;LS&gt;</td>
<td></td>
<td>Multicast GroupID</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# IPv6 – The Basics

## Address Type Rehash

<table>
<thead>
<tr>
<th>Type</th>
<th>Range</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Unicast</td>
<td>2000::/3</td>
<td>Normal host to host communications</td>
</tr>
<tr>
<td>Link-local</td>
<td>fe80::/10</td>
<td>Connected-link communications</td>
</tr>
<tr>
<td>Anycast</td>
<td>2000::/3</td>
<td>One to Any (Any=topologically closest)</td>
</tr>
<tr>
<td>Unique-local</td>
<td>fc00::/7</td>
<td>Normal host to host communications within a site</td>
</tr>
<tr>
<td>IPv4-mapped</td>
<td></td>
<td>Represent an IPv4 address in IPv6 format</td>
</tr>
<tr>
<td>6to4</td>
<td>2002::/3</td>
<td>Automatic transition mechanism</td>
</tr>
<tr>
<td>ISATAP</td>
<td></td>
<td>Intra-site transition mechanism</td>
</tr>
<tr>
<td>Unspecified</td>
<td>::/0</td>
<td>When node doesn’t have an address</td>
</tr>
<tr>
<td>Loopback</td>
<td>::1/128</td>
<td>Used for intra-node communications</td>
</tr>
<tr>
<td>Multicast</td>
<td>ff00::/8</td>
<td>One to Many</td>
</tr>
</tbody>
</table>
IPv6 -- The basics
How Global Unicast is Allocated

- 2000::/3
- 0::/0 (IETF » IANA)
- 2610::/12
- 261f:1::/32 (204 /32s per Pixel)
- 261f:1:d405::/48 (409.6 /48s per pixel)

IANA or RIR » End Site
IPv6 -- The basics
How Global Unicast is Allocated

The Numbers:
- 8 /3s, one of which is in use
- 512 /12 allocations to RIRs in first /3 (6 used so far)
- 1,048,576 LIR /32s in each RIR /12
- 65,536 /48 Assignments in each /32

261f:1:d405::/48 (409.6 /48s per pixel)

(IANA or RIR » End Site)

261f:1:d405:e008:/48 (409.6 /64s per pixel)

(IANA or RIR » End Site)
IPv6 -- The Basics

Global Unicast in perspective

- The Numbers (cont.)
  - The first /12 assigned to each RIR can support 68,719,476,736 /48 End Sites
  - There are 506 /12s remaining if that’s not enough for any particular region.
  - Many ISPs will require more than a /32, but, even if we figure a /28 for every ISP on average, that’s still enough addresses for 65,536 ISPs in each RIR region without exhausting their first /12. (There are currently fewer than 30,000 BGP speaking ISPs worldwide)
  - In short... There is more than enough address space for liberal assignments under current and any likely policy.
IPv6 -- Not your father’s IP

THE FUTURE
Ain’t what it used to be.

motifake.com
IPv6 -- Address Planning
Don’t oversimplify too much!

- There are lots of people saying “ISPs get /32s, end sites get /48s.”
- That’s an unfortunate oversimplification.
- ISPs get AT LEAST a /32 and can get whatever larger allocation they can justify.
- End sites should get at least a /48 and should be given whatever larger assignment they can justify.
IPv6 -- Address Planning Methodology

- Don’t start with a /32 and figure out how to make your needs fit within it.
- Start by analyzing your needs and apply for a prefix that will meet those needs.
- In your analysis, it’s worth while to try and align allocation units to nibble boundaries. A nibble boundary is a single hex digit, or, a number $2^n$ such that $n$ is a multiple of 4. (e.g. 16, 256, 4096, 16384, 65536...)
IPv6 Address Planning Analysis

- Start with the number of end sites served by your largest POP. Figure a /48 for each. Round up to the a nibble boundary. (if it’s 3,000 end sites, round up to 4096, for example... a /36 per POP.

- Next, calculate the number of POPs you will have. Include existing POPs and likely expansion for several years. Round that up to a nibble boundary, too. (140 POPs, round up to 256).
IPv6 Address Planning Analysis

- Now that you have an address size for each POP (4096 = 12 bits in our example) and a number of POPs (256 = 8 bits in our example), you know that you need a total of POP*nPOPs /48s for your network (4096*256=1,048,576 or 12+8=20 bits).
- 48 bits - 20 bits is 28 bits, so, you actually need a /28 to properly number your network.
- You probably could squeeze this into a /32, but, why complicate your life unnecessarily?
IPv6 Address Planning
Apply for your addresses

- Now that you know what size block you need, the next step is to contact your friendly neighborhood RIR (Regional Internet Registry) and apply.

- Most RIRs provide either an email-based template or a web-based template for you to fill out to get addresses.

- If you are a single-homed end-user, you usually should get your addresses from your upstream rather than an RIR.
IPv6 Address Planning

The bad news

- The addressing methodology I described above may not be consistent with RIR policy in all regions (yet)*.
- This means you might have to negotiate to a smaller block.
- All RIRs have an open policy process, so, you can submit a proposal to enable this kind of allocation, but, that may not help you immediately.

* Prop-098 under discussion in APNIC, Adopted 2011-3 in ARIN, mostly permitted in RIPE, not yet discussed in AfriNIC or LACNIC.
IPv6 Address Planning
The good news

- Having things on nibble boundaries is convenient, but, not necessary.
  - ip6.arpa DNS delegations
  - Human Factors
  - Routing Table management
  - Prefix lists

- The techniques that follow should work either way.
IP Address Planning  
Carving it up

- For the most part, you’ve already done this.
- Take the number you came up with for the nPOP round-up and convert that to a number of bits (256 = 8 bits in our example).
- Now, take what the RIR gave you (/28 in our example) and add that number to the above number (28+8 = 36) and that’s what you need for each POP (a /36 in our example).
IPv6 Address Planning

Carving it up

- Now let’s give address segments to our POPs.
- First, let’s reserve the first /48 for our infrastructure. Let’s use 2000:db80 - 2000:db8f as our example /28.
- Since each POP gets a /36, that means we have 2 hex digits that designate a particular POP.
- Unfortunately, in our example, that will be the last digit of the second group and the first digit of the third group.
IPv6 Addressing
Carving it up

Strategy

- Sequential Allocation
  - Advantage: Simple, easy to follow
  - Advantage: POP Numbers correspond to addresses
  - DisAdvantage: Complicates unexpected growth

- Allocation by Bisection
  - Advantage: Simplifies growth
  - Advantage: Greatest probability of Aggregation
  - Disadvantage: “Math is hard. Let’s go shopping!”
IPv6 Addressing
Allocation by Bisection

- Bisection? What does THAT mean?
- Simple... It means to cut up the pieces by taking the largest remaining piece and cutting in half until you have the number of pieces you need.
- Imagine cutting up a pie into 8 pieces...
IPv6 Addressing Allocation by Bisection

- Bisection? What does THAT mean?
- Simple... It means to cut up the pieces by taking the largest remaining piece and cutting in half until you have the number of pieces you need.
- Imagine cutting up a pie into 8 pieces...

First, we cut it in half...
IPv6 Addressing Allocation by Bisection

- Bisection? What does THAT mean?
- Simple... It means to cut up the pieces by taking the largest remaining piece and cutting in half until you have the number of pieces you need.
- Imagine cutting up a pie into 8 pieces...

Then we cut it in half again
IPv6 Addressing
Allocation by Bisection

- Bisection? What does THAT mean?
- Simple... It means to cut up the pieces by taking the largest remaining piece and cutting in half until you have the number of pieces you need.
- Imagine cutting up a pie into 8 pieces...
IPv6 Addressing Allocation by Bisection

- Bisection? What does THAT mean?
- Simple... It means to cut up the pieces by taking the largest remaining piece and cutting in half until you have the number of pieces you need.
- Imagine cutting up a pie into 8 pieces...

And finally a fourth cut
IPv6 Addressing
Allocation by Bisection

- It’s a similar process for IPv6 addresses.
  - Let’s start with our 2001:db80::/28 prefix.
  - We’ve already allocated 2001:db80:0000::/48
IPv6 Address Planning Allocation by Bisection

- After repeating this for 19 POP allocations, we have a table that looks like this:

<table>
<thead>
<tr>
<th>Infrastructure</th>
<th>2001:db80:0000::/48</th>
<th>POP1</th>
<th>2001:db88:0000::/36</th>
</tr>
</thead>
<tbody>
<tr>
<td>POP8</td>
<td>2001:db81:0000::/36</td>
<td>POP9</td>
<td>2001:db89:0000::/36</td>
</tr>
<tr>
<td>POP4</td>
<td>2001:db82:0000::/36</td>
<td>POP5</td>
<td>2001:db8a:0000::/36</td>
</tr>
<tr>
<td>POP2</td>
<td>2001:db84:0000::/36</td>
<td>POP3</td>
<td>2001:db8c:0000::/36</td>
</tr>
<tr>
<td>POP6</td>
<td>2001:db86:0000::/36</td>
<td>POP7</td>
<td>2001:db8e:0000::/36</td>
</tr>
<tr>
<td>POP18</td>
<td>2001:db87:0000::/36</td>
<td>POP19</td>
<td>2001:db8f:0000::/36</td>
</tr>
</tbody>
</table>
IPv6 Address Planning
Allocation by Bisection

- Notice how by doing that, most of the /36s we created have 15 more /36s before they run into allocated space and all have at least 7.

- Notice also that if any POPs get larger than we expect, we can expand them to /35s, /34s, /33s, and most all the way to a /32 without having to renumber.

- By default, at /36, each pop has room for 4096 /48 customers. End sites that need more than a /48 should be extremely rare*. 

IPv6 Address Planning
Allocation by Bisection

* End Site means a single customer location, not a single customer. Many customers may need more than a /48, but, with 65,536 /64 subnets available, even the largest building should be addressable within a /48.
TAKE A BREAK

- It’s come time to accommodate that basic reality... Humans don’t do well sitting in one place for extended periods of time listening to the same person blather on and on no matter how interesting the topic.
- EVERYONE stand up.
- EVERYONE leave the room for at least the next 10 minutes.
- EVERYONE try to be back within 20 minutes, please. There’s still lots to cover.
This is the Internet

This is the Internet on IPv4 (2012)

Any questions?
Quick Interactive Review

- OK... Let’s see what you remember from the first half:
Quick Interactive Review

- OK... Let’s see what you remember from the first half:
- Plan a small ISP:
  - 300 End sites per POP
  - 12 POPs
Quick Interactive Review

- OK... Let's see what you remember from the first half:
- Plan a small ISP:
  - 300 End sites per POP
  - 12 POPs
- Plan a medium ISP:
  - 500 End sites per POP
  - 150 POPs
Quick Interactive Review

- OK... Let’s see what you remember from the first half:
  - Plan a small ISP:
    - 300 End sites per POP
    - 12 POPs
  - Plan a medium ISP:
    - 500 End sites per POP
    - 150 POPs
  - Plan a very large ISP:
    - 1000 End sites per POP
    - 800 POPs
LoL Kitteh sez:

More IPv4 NAT
Are you fscking kidding me?
Part 2 – Making it work

Now that we’ve covered the nuts and bolts, let’s talk about how they go together into subassemblies and assemblies to make a working network.
IPv6 Host Administration

- Configuring IPv6 on Linux
  - `/etc/network/interfaces`
  - How to configure SLAAC
  - How to configure a Static Address
IPv6 and DNS

- Forward IPv6 resolution
  - AAAA Records
    - Mostly the same as A records
    - Right Hand side is an IPv6 Address in standard abbreviated form (e.g. 2620:0:930::200:2)
    - 96 more bits, no magic.

- Reverse IPv6 resolution
  - PTR Records, just like IPv4
    - ip6.arpa instead of in-addr.arpa
    - No abbreviating
    - Each nibble is a zone boundary
IPv6 DNS Examples

- **Forward:**
  
  dualhost.example.com. IN AAAA 2001:db8::5
  
  IN A 192.9.200.5

- **Reverse:**
  
  5.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. IN PTR ipv6host.example.com
  
  or:
  
  $ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
  
  5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
  
  IN PTR dualhost.example.com.

- **Note:** in-addr.arpa and ip6.arpa are separate zones maintained in separate files.
IPv6 Deployment

Native connectivity

- Native connectivity is, by far, the simplest and cleanest way to deploy IPv6.
- Just like IPv4 (mostly) but with bigger addresses that look different.

Example:

```
DEVICE=eth0
ONBOOT=yes
MTU=1280
IPADDR=192.159.10.2
NETMASK=255.255.255.0
GATEWAY=192.159.10.254

IPV6INIT=yes
IPV6ADDR=2620:0:930::0200:1/64
IPV6_DEFAULTGW=2620:0:930::1
IPV6_AUTOCONF=no
```
IPv6 Deployment Using Tunnels

- Slightly more complicated than native connectivity
- MTU problems
- Lower MTU reduces performance on some platforms. (Significantly on some versions of Windows)
- Harder to Configure or harder to Troubleshoot (depending on tunnel solution chosen)
IPv6 Tunneling

6in4 or GRE Tunnels

- Require manual configuration
- Direct point to point tunnel between two known (and configured) IPv4 nodes
- 6in4 is very similar to GRE, but, only IPv6 inside IPv4 payload, protocol 41.
- GRE can support both IPv6 and IPv4 in an IPv4 payload, protocol 47.

Example:
gr-0/0/0 {
  unit 2 {
    description "HE Tunnel Broker";
    point-to-point;
    tunnel {
      source 24.4.178.41;
      destination 64.71.128.83;
      path-mtu-discovery;
    }
    family inet6 {
      mtu 1280;
      address 2001:470:1f03:9c::2/64;
    }
  }
}

Monday, October 31, 2011
IPv6 Tunneling

6to4 Autotunneling

- Also uses protocol 41 like 6in4 tunneling, but, uses IPv4 anycast and automatic IPv6 address construction based on IPv4 address.
- Nothing to configure, per se, just turn it on.
- Non-deterministic.
- Changes can occur anywhere in the network.
- No control over which 6to4 gateway you use.
- Hard to troubleshoot
- Easy to deploy (when it works)
IPv6 tunneling
Teredo

- Developed by Microsoft.
- Automatically on by default in Windows since Vista
- Present in Windows since XP SP1
- Works through NAT
- Even Microsoft calls it a “Last Resort” for IPv6 connectivity. Unless you absolutely need it, disable it and block it at your firewall.
IPv6 Routing
RIPng

- Just like RIPv2 for IPv4
- Don’t do this
- Same reasons as RIPv2.
- Don’t do this
- It’s just a bad idea.
- Really, Don’t use RIP.
IPv6 Routing

OSPF

- OSPF3 can handle both IPv6 and IPv4 (most vendors)
- OSPF2 (common version) is only IPv4.
- OSPF3 available on most routers, but, requires configuration as OSPF3, not just “OSPF”.
- Can run OSPF2 for IPv4 and OSPF3 simultaneously on most routers
- Largely like IPv4 OSPF2 configuration
IPv6 Routing
OSPFv3 differences from OSPFv2

- Router and Network LSA don’t carry IP addresses, new type 9 LSA instead.
- Relies on AH and ESP instead of having authentication in OSPF protocol
- Advertises all prefixes on the interface
- LSAs now have Flooding Scope
- Supports multiple instances per link - only routers in the same instance form adjacencies
- For all neighbor communications, all packets are sourced from link-local addresses with the exception of virtual links.
## IPv6 Routing

### OSPFV3 LSA types

<table>
<thead>
<tr>
<th>LSA Type</th>
<th>Common Name</th>
<th>Description</th>
<th>Flooding Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Router LSA</td>
<td>Describes a router's link states and costs of its links to one area.</td>
<td>Area</td>
</tr>
<tr>
<td>2</td>
<td>Network LSA</td>
<td>Generated by a DR to describe the aggregated link state and costs for all routers attached to an area.</td>
<td>Area</td>
</tr>
<tr>
<td>3</td>
<td>Inter-Area Prefix LSA for ABRs</td>
<td>Originated by ABRs to describe interarea networks to routers in other areas.</td>
<td>Area</td>
</tr>
<tr>
<td>4</td>
<td>Inter-Area Router LSA for ASBRs</td>
<td>Originated by ASBRs to advertise the ASBR location.</td>
<td>Area</td>
</tr>
<tr>
<td>5</td>
<td>Autonomous System External LSA</td>
<td>Originated by an ASBR to describe networks learned from other protocols (redistributed routes).</td>
<td>Autonomous System</td>
</tr>
<tr>
<td>8</td>
<td>Link LSA</td>
<td>Advertises link-local address and prefix(es) of a router to all other routers on the link as well as option information. Sent only if more than one router is present on a link.</td>
<td>Link</td>
</tr>
<tr>
<td>9</td>
<td>Inter-Area Prefix LSA for ABRs</td>
<td>Performs one of two functions:</td>
<td>Area</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Associates a list of IPv6 prefixes with a transit network by pointing to a Network LSA.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Associates a list of IPv6 prefixes with a router by pointing to a Router LSA.</td>
<td></td>
</tr>
</tbody>
</table>
IPv6 Routing

OSPFv3 Configuration

- Still requires a 32 bit Router ID
  - Either configure a loopback interface with an IPv4 address or
  - Set the router-id in the appropriate block
  - Note: router-id does not require an IPv4 address and OSPFv3 can work without IPv4.

- Example: Juniper SRX-100

```
ospf3 {
  export static-to-ospf;
  area 0.0.0.0 {
    interface lo0.0 {
      passive;
    }
    interface fe-0/0/0.0;
    interface gr-0/0/0.20;
    interface gr-0/0/0.21;
  }
}
```
IPv6 Routing

eBGP

- Nearly identical to BGP for IPv4
- Different vendors have different ways of differentiating routing tables
  - Cisco: address-family ipv6
    - Misconfigurations usually silently ignore your intent
  - Juniper: family inet6
    - Misconfigurations usually result in an error message
IPv6 Routing

eBGP

- Configuration Example:
  - Juniper SRX-100

```c
bgp {
  family inet {
    unicast;
  }
  family inet6 {
    unicast;
  }
  local-as 1734;
  group l42 {
    family inet {
      unicast;
    }
    family inet6 {
      unicast;
    }
    export to-l42;
    peer-as 8121;
    neighbor 192.124.40.129;
    neighbor 2620::930:7ffe::1 {
      family inet6 {
        unicast;
      }
    }
  }
}
```
IPv6 Routing
iBGP

- Same set of differences as IPv4 e vs. i BGP.
- All border routers must peer in a mesh.
- IGP must provide reachable routes to all routers.
- Peering best between loopbacks.
- All the IPv4 lessons and best practices apply.
IPv6 Routing
iBGP

- Configuration Example:
  - Juniper SRX-100

```plaintext
bgp {
    local-as 1734;
    group internal {
        type internal;
        export ibgp-next-hop-self;
        peer-as 1734;
        neighbor 2620:0:930:7000::1 {
            family inet6 {
                unicast;
            }
        export v6-next-hop-self;
        }
    neighbor 192.124.40.193 {
        local-address 192.124.40.194;
        import via-cable;
        family inet {
            unicast;
        }
    }
    }
    }
    ```
IPv6 -- Much like IPv4 was supposed to be

- This time with enough Addresses
  - Basic weight comparison: If IP addresses were a unit of mass, then, if IPv4 weighed the same as 7 liters of water, IPv6 would weigh the same as the entire planet Earth.
IPv6 Operations

DHCPv6 Overview

- Can assign prefix lengths other than /64
- Bootstrapped by RS/RA process
- DHCP-PD -- Potentially very useful feature for ISPs
- Can provide DNS, NTP, and other server locations (RA/SLAAC cannot, but, ND has a process for doing DNS servers).
- Limited field deployment/support so far.
- IETF religious problem (SLAAC vs. DHCP war)
IPv6 Operations
SLAAC and the need for RA–Guard

- Stateless Autoconfiguration provides a very convenient way to number hosts.
- Unfortunately, all routers are created equal, including the ones created by someone else.
- In DHCP, a rogue server tends to break things.
- With RA, a rogue router can intercept traffic without breaking anything.
- RA Guard blocks RA transmissions on switchports not defined as routers.
IPv6 Transition Issues
Things that don’t route packets

- **Software Upgrades**
  - In-house tools
    - Porting to IPv6 network
    - Adding IPv6 address capability to
      - databases
      - parsers
      - data structures
      - etc.
  - Vendor Provided Software
    - Start talking to them now if you aren’t already
    - You’re not the only one asking, no matter what they say.
    - Call their bluff.
IPv6 Transition Issues
Things that forward packets

- Hardware Upgrades
  - Firewalls
  - Really old routers
  - Intrusion Detection systems
  - Load Balancers
  - Printers -- Probably just leave on internal IPv4.
  - Clock sources
  - Other infrastructure
  - Other odd appliances
Parting Thoughts: What if IP Addresses were M&Ms -- IPv4

- IPv4: Standard network size: /24
  - Number of usable addresses: 254
  - Number of /24s: Aprox. 14.5 Million

One IPv4 /24 -- 254 M&Ms

Full Address Space, One M&M per /24 covers 70% of a football* field

*An American Football field
Parting Thoughts: IP Addresses as M&Ms -- IPv6

- How many M&Ms in IPv6?
  - Standard Network size: /64
    - Host addresses: 18,446,744,073,709,551,616
    - Number of Networks: 18,446,744,073,709,551,616

- In short, Enough M&Ms to fill the great lakes in either measure*

*Warning: Do not eat 18,446,744,073,709,551,616 M&Ms as adverse health consequences are likely.
If IP addresses were M&Ms
Final thought:
If IP addresses were M&Ms
Final thought:

They’d taste better.
OK.. The real conclusion

- You’ve learned a lot about the structure of IPv6 and IPv6 addressing.
- You’ve learned how to plan your addressing strategy.
- You’ve learned a little about deploying IPv6.
- Mostly, deployment is a lot like IPv4, but, without the address scarcity.
Similarities

- Same routing protocols (mostly)
- Same issues
- Same solutions (mostly)
- Similar security models and tactics
  - Note: NAT isn’t security in IPv4. Stateful inspection provides security, NAT just depends on stateful inspection.
- Similar host configuration methods
Differences

- Much bigger address space
- Restoration of the End-to-End model of networking (Really, this is a good thing, even if you find it a little scary at first)
- Stateless Autoconfiguration
- (Slightly) Crippled DHCP
- DHCP Prefix Delegation
- Hex Notation
- No more writing out netmasks (YAY!)
IPv6 Anycast

Two Kinds of Anycast

- Both essentially provide a common service from multiple hosts.
- Both deliver a packet to at least one of a group of hosts sharing a common IP address.
- IPv6 RFCs define certain types of subnet local anycast addresses (prefix:fdff:ffff:ffff:ffxx where xx is in the range 80-ff)
- More general anycast is implemented identical to IPv4 Anycast and is in much more common use.
IPv6 Deployment -- Next Steps

- This was just an introduction.
- Hopefully you have enough knowledge to build an IPv6 test lab and start gaining experience.
- Use a tunnel at first if you need to.
- Don’t forget to pressure your vendors so that they are ready when you need them to be (or only a little late)
  - Network
  - Equipment
IPv6 Deployment -- Next Steps part 2

- Build your test lab
- Try it out
- Try different scenarios
- Get to know what works and doesn’t work in your particular environment

Help is available
- ARIN IPv6 wiki: http://www.getipv6.info
- Hurricane Electric: http://www.tunnelbroker.net
- “Rent” me: owend@he.net
Q&A

Contact:

Owen DeLong
IPv6 Evangelist
Hurricane Electric
760 Mission Court
Fremont, CA 94539, USA
http://he.net/
owend at he dot net
+1 (408) 890 7992
The end

Thank you

Contact:

Owen DeLong
IPv6 Evangelist
Hurricane Electric
760 Mission Court
Fremont, CA 94539, USA
http://he.net/

owend at he dot net
+1 (408) 890 7992