NAP Algorithm performances and evolutions

Laurent Toutain, David Binet, Eric Fleury, Guillaume Chelius, Bruno Sericola.

Laurent.Toutain@enst-bretagne.fr
David.Binet@orange-ftgroup.com
Home Networking Definition

• Millions of users:
  • Relatively low number of equipments
  • But complex and arbitrary topology:
    • L2 links supporting IPv6

• Competition between providers:
  • Multi-homing
  • No collaboration between providers

• Reduce Provider management costs
• No configuration from the user
• **User can build complex architecture**
  • If Bridging is used: loops must be detected
  • Spanning Tree is not efficient for Traffic Engineering
  • Traffic will converge on some links

• **Routing will allow more control:**
  • Routers have to be configured

\[
\begin{array}{c|c|c}
GP &= \text{provider} & \text{SID} &= ? & I-ID &= \text{autoconf} \\
\end{array}
\]
DHCPv6 Prefix Delegation

**Main idea: The edge router**
- become the DHCPv6 server for prefixes (/64) for the home network.
- Get a global prefix for the provider.
- Create a pool of GP:SID to reach the /64 boundary
- Allocate these prefixes to routers

**When a router starts**
- Periodically broadcast requests until receiving an answer from a DHCPv6 server
- When configured act as a DHCPv6 relay.

**More studies on multi-homing and network stability are needed**
No Administration Protocol

- draft-chelius-router-autoconf-00.txt

Main idea:
- GP is flooded to routers.
- One router on a link select randomly a SID to reach /64 limit.
- SID is flooded to detect collision.
- If collision smallest Router ID keep it, other routers drawn another value amongst the available ones.
- When there is no conflict on SID value, Prefix (/64) is announced through Neighbor Discovery.

Deal with multi-homing (either is GP length different)
- SID will be different for each prefix on a link

Algorithm efficiency?
Synchronous Algorithm

\[ NCL : \text{Non Configured Links} \]

while \((NCL \neq \emptyset)\) do
  for \((i \in NCL)\) do
    \(v(i) = \text{random(SID)};\)
    Broadcast;
  for \((i \in NCL \text{ such that } v(i) \in \text{SID})\) do
    \(NCL = NCL - \{i\};\)
    \(\text{SID} = \text{SID} - \{v(i)\};\)
\end{align*}

Input: \(M\) urns and \(L\) balls
Pre condition: \(M \geq L\)

if \((L \neq 0)\) then
  Randomly throw the \(L\) balls into the \(M\) urns;
  \begin{enumerate}
    \item Let \(c \leq M\) denote the number of urns containing at least one ball;
    \item Keep aside these \(c\) urns and for each of them one of the balls contained inside;
    \item Call NAP Process\((M - c, L - c)\);
  \end{enumerate}
end
model (continued)

- \( P_{i,j} \): Probability to obtain exactly \( j-i \) non empty urns when throwing \( L-i \) balls in \( M-i \) urns:

\[
p_{i,j}(L, M) = \binom{M - i}{j - i} \sum_{k=0}^{j-i} \binom{j - i}{k} (-1)^k \left( \frac{j - i - k}{M - i} \right)^{L-i}
\]

- Mean absorption time \( V_0 \) is given by:

Input: \( M \) urns and \( L \) balls
Pre condition: \( M \geq L \geq 1 \)
\( V_{L-1} = 1 \);
for \( i = L - 2 \) downto 0 do
  \[
  V_i = 1 + \sum_{j=i+1}^{L-1} p_{i,j}(L, M)V_j;
  \]
end
### Results

<table>
<thead>
<tr>
<th>GP vs #Links</th>
<th>5 links</th>
<th>10 links</th>
<th>15 links</th>
<th>20 links</th>
<th>25 links</th>
</tr>
</thead>
<tbody>
<tr>
<td>/60</td>
<td>1.5</td>
<td>2.19</td>
<td>3.06</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>/56</td>
<td>1.04</td>
<td>1.16</td>
<td>1.34</td>
<td>1.53</td>
<td>1.70</td>
</tr>
<tr>
<td>/52</td>
<td>1.00</td>
<td>1.01</td>
<td>1.03</td>
<td>1.05</td>
<td>1.07</td>
</tr>
<tr>
<td>/48</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

*Fast convergence*
### Results

<table>
<thead>
<tr>
<th>GP vs #Links</th>
<th>5 links</th>
<th>10 links</th>
<th>15 links</th>
<th>20 links</th>
<th>25 links</th>
</tr>
</thead>
<tbody>
<tr>
<td>/60</td>
<td>1.5</td>
<td>2.19</td>
<td>3.06</td>
<td>n.a.</td>
<td>n.a.</td>
</tr>
<tr>
<td>/56</td>
<td>1.04</td>
<td>1.16</td>
<td>1.34</td>
<td>1.53</td>
<td>1.70</td>
</tr>
<tr>
<td>/52</td>
<td>1.00</td>
<td>1.01</td>
<td>1.03</td>
<td>1.05</td>
<td>1.07</td>
</tr>
<tr>
<td>/48</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

*No conflicts*

Fast convergence
Implementation in OSPF

- Use OSPFv3 properties:
  - Database synchronization, 3 opaque types
    - GP LSA: flooded GP to every routers on AS
    - SID proposed: flooded to every routers on area to detect collisions
    - SID assigned: flooded on the link to inform other routers of the selected prefix (GP:SID)
  - DR election
    - Only DR on each link participate to the SID selection
  - Selected prefixes are injected in the topology database for routing
Partitioning and merging

- Differentiate link failure and router failure
Partitioning and merging

- Differentiate link failure and router failure
Partitioning and merging

- Differentiate link failure and router failure
Partitioning and merging

- Differentiate link failure and router failure
Partitioning and merging

- Differentiate link failure and router failure
Partitioning and merging

- Differentiate link failure and router failure

![Diagram showing partitioning and merging with SID1 and DR]
Partitioning and merging

- Differentiate link failure and router failure
Partitioning and merging

- Differentiate link failure and router failure
Partitioning and merging

- Differentiate link failure and router failure

Diagram:
- SID1
- DR
- GP:SID1
Partitioning and merging

- Differentiate link failure and router failure
Partitioning and merging

- Differentiate link failure and router failure
Partitioning and merging

• Differentiate link failure and router failure

![Diagram showing SID1 and SID2 with DR (Designated Router) and GP (Global Prefix)]
Partitioning and merging

- Differentiate link failure and router failure
Partitioning and merging

- Differentiate link failure and router failure
Partitioning and merging

- Differentiate link failure and router failure

![Diagram showing SID1 and SID2 with DR and GP:SID1]
Distant Vector?

- **OSPF has some drawbacks:**
  - Router ID (32 bits) too short to guaranty uniqueness
  - Too complex (memory, processor) for a router between a bluetooth and PLC network

- **Distant Vector are known to be simpler to implement:**
  - DV can be used to propagate GP
  - SID allocated \(\subset\) FIB + Neighbor Discovery to inform other routers on the link
  - SID proposed: use DV flooding + flags (don’t install in the FIB, collision detected)
  - “Designated Router” election: to be implemented
- For NAP algorithm, prefix must be flooded on the site
  - change forwarding behavior for default route:
    ➡ When a default route is selected:
      ➡ source address is a Global Address
      ➡ have different entries in routing table for each GP
      ➡ longest prefix match on the source address
      ➡ DV cost will avoid loops
Multi-homing routing

- For NAP algorithm, prefix must be flooded on the site
  - change forwarding behavior for default route:
    ➡️ When a default route is selected:
      ➡️ source address is a Global Address
      ➡️ have different entries in routing table for each GP
      ➡️ longest prefix match on the source address
      ➡️ DV cost will avoid loops
Multi-homing routing

- For NAP algorithm, prefix must be flooded on the site
  - change forwarding behavior for default route:
    - When a default route is selected:
      - source address is a Global Address
      - have different entries in routing table for each GP
      - longest prefix match on the source address
      - DV cost will avoid loops
Conclusion

- NAP algorithm has been proven to be efficient.
- Compatible with CIDR routing, keep same interaction with the provider.
- Compatible with shim6 : applications can select the source address and the provider.
- A new family of routing protocol can be developed for stub networks (SME, Home,...).
  - Releasing the scalability constraint allow auto-configuration and multi-homing routing