batman-adv scalability
Layer 2 Mesh Networks - Myths and Risks

Linus Lüssing

Freifunk Hamburg Geekend02, Sep. 2013
Outline

1. Introduction
   - Layer 2 Mesh Networks

2. Past
   - Experiences From Lübeck

3. Present
   - Statistics From Hamburg

4. Future
   - Features in Development
Outline

1. Introduction
   - Layer 2 Mesh Networks

2. Past
   - Experiences From Lübeck

3. Present
   - Statistics From Hamburg

4. Future
   - Features in Development
batman-adv: Big, Virtual Switch
batman-adv: Big, Virtual Switch
batman-adv: Encapsulation

- Encapsulates ethernet frames
- In own batman-adv header
Advantages

- Supports IPv4, IPv6, probably IPv42, ...
- Your non-IP / link-layer protocol?
- More flexible than Linux IP routing table:
  - Interface bonding
  - Network coding
  - ...
- Simple configuration
  - MAC addresses are unique
  - No IP subnet coordination
- Fast Roaming
- ...
Disadvantage: Overhead

Layer 3 mesh routing protocol:
- Mesh protocol overhead

Layer 2 mesh routing protocol:
- Mesh protocol overhead
- + Layer 2 specific overhead
Disadvantage: Overhead

IEEE 802.11s:
- Designed for ~32 nodes
Layer 2 Mesh Networks

Disadvantage: Overhead

"Layer 2 Mesh Networks? Don’t Scale!"
Outline

1. Introduction
   - Layer 2 Mesh Networks

2. Past
   - Experiences From Lübeck

3. Present
   - Statistics From Hamburg

4. Future
   - Features in Development
Experiences from Freifunk Lübeck
1 Node: Kernel panics

- 2009, pre 0.1 firmware
- Did not boot: batman-adv crashing
1 Node: Kernel panics

- 2009, pre 0.1 firmware
- Did not boot: batman-adv crashing
Trying out batman-adv over VPN, using tinc:

- tinc does meshing, too:
- Large neighbourhood on VPN: many rebroadcasts
- Overhead on DSL got close to 1MBit/s
Changes in 0.2

October 2011

- Added tinc
- OGM (= route update + link quality) interval:
  - 1s -> 3s
- Should scale to 30 nodes then, right?
15-20 Nodes: Again, Too Large Neighborhood on VPN

Wrong assumption about OGM scalability:
- Linear to number of nodes in line topology
- But squared to number of local neighbour nodes
Changes in 0.3

August 2012: fastd

- Decreased neighbourhood size on VPN
- Peer-Groups: Connection to two of n gateway nodes
80 Nodes: Multicast Overhead

- Two wifi neighbours
- Measured on adhoc wifi interface
- Result: Losing about 25% airspace
80 Nodes: Multicast Overhead - Types

- Service Announcements: SSDP, LLMNR, MDNS
- Address Resolution: ARP+ICMPv6
April, 2013

- batman-adv 2013.0.0: Distributed ARP Table
- Multicast Rate: 1MBit/s $\Rightarrow$ 12MBit/s
- Rebroadcasts on VPN: 3 $\Rightarrow$ 1
- Filter for non-essential multicast packets
Outline

1. Introduction
   - Layer 2 Mesh Networks

2. Past
   - Experiences From Lübeck

3. Present
   - Statistics From Hamburg

4. Future
   - Features in Development
Test Setup

- tcpdump on fastd VPN tunnel interface
- Thu Sep 19 00:00:00 2013 - Thu Sep 19 23:59:59 2013
RX by batman-adv type, average Bits/s

- fastd: +66Bytes headers
- ~ 94KBit/s average
- 1.5% of ADSL, 6MBit/s
TX by batman-adv type, average Bits/s

- ~ 80KBit/s average
- 14% of ADSL, 576KBit/s
RX by batman-adv type, Packets/180s

- OGM $\propto \#\text{clients}$
- BCAST $\propto \#\text{clients}$
RX by multicast type, average Bits/s

- Bandwidth hog: ICMPv6
- Many nodes with old firmware:
  - DAT not running ideal
  - Still much: SSDP+MDNS+NBNS+LLMNR
- Size vs. Rate
RX by ICMPv6 type, average Bits/s

Most: Neighbor Discovery
OGM and layer 2 specific multicast overhead about the same
IPv6 ND is currently the largest layer 2 specific overhead
Getting close to the ADSL upload limit
Outline

1. Introduction
   - Layer 2 Mesh Networks

2. Past
   - Experiences From Lübeck

3. Present
   - Statistics From Hamburg

4. Future
   - Features in Development
"Split horizon" for multicast payload frames

- No rebroadcasts for packets on/from VPN interface
"Split horizon" for multicast payload frames

Thu Sep 26 21:14:08 UTC 2013
eth1 / traffic statistics

<table>
<thead>
<tr>
<th></th>
<th>rx</th>
<th>tx</th>
</tr>
</thead>
<tbody>
<tr>
<td>bytes</td>
<td>19.83 MiB</td>
<td>10.39 MiB</td>
</tr>
<tr>
<td>max</td>
<td>572 kbit/s</td>
<td>340 kbit/s</td>
</tr>
<tr>
<td>average</td>
<td>90.25 kbit/s</td>
<td>47.31 kbit/s</td>
</tr>
<tr>
<td>min</td>
<td>32 kbit/s</td>
<td>16 kbit/s</td>
</tr>
<tr>
<td>packets</td>
<td>223391</td>
<td>102620</td>
</tr>
<tr>
<td>max</td>
<td>758 p/s</td>
<td>395 p/s</td>
</tr>
<tr>
<td>average</td>
<td>124 p/s</td>
<td>57 p/s</td>
</tr>
<tr>
<td>min</td>
<td>38 p/s</td>
<td>16 p/s</td>
</tr>
<tr>
<td>time</td>
<td>30.00 minutes</td>
<td></td>
</tr>
</tbody>
</table>

Thu Sep 26 21:44:08 UTC 2013
### Features in Development

**“Split horizon” for multicast payload frames**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>rx</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bytes</td>
<td>19.59 MiB</td>
<td>19.59 MiB</td>
</tr>
<tr>
<td>max</td>
<td>488 Kbit/s</td>
<td>488 Kbit/s</td>
</tr>
<tr>
<td>average</td>
<td>89.16 Kbit/s</td>
<td>89.16 Kbit/s</td>
</tr>
<tr>
<td>min</td>
<td>32 Kbit/s</td>
<td>32 Kbit/s</td>
</tr>
<tr>
<td>packets</td>
<td>229231</td>
<td>229231</td>
</tr>
<tr>
<td>max</td>
<td>635 p/s</td>
<td>635 p/s</td>
</tr>
<tr>
<td>average</td>
<td>123 p/s</td>
<td>123 p/s</td>
</tr>
<tr>
<td>min</td>
<td>45 p/s</td>
<td>45 p/s</td>
</tr>
<tr>
<td>time</td>
<td>30.00 minutes</td>
<td>30.00 minutes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>tx</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bytes</td>
<td>3.24 MiB</td>
<td>3.24 MiB</td>
</tr>
<tr>
<td>max</td>
<td>56 Kbit/s</td>
<td>56 Kbit/s</td>
</tr>
<tr>
<td>average</td>
<td>14.73 Kbit/s</td>
<td>14.73 Kbit/s</td>
</tr>
<tr>
<td>min</td>
<td>7 p/s</td>
<td>7 p/s</td>
</tr>
<tr>
<td>packets</td>
<td>28916</td>
<td>28916</td>
</tr>
<tr>
<td>max</td>
<td>16 p/s</td>
<td>16 p/s</td>
</tr>
<tr>
<td>average</td>
<td>4 p/s</td>
<td>4 p/s</td>
</tr>
<tr>
<td>min</td>
<td>7 p/s</td>
<td>7 p/s</td>
</tr>
</tbody>
</table>

Thu Sep 26 21:14:06 UTC 2013

---

**eth2 / traffic statistics**

Thu Sep 26 21:44:06 UTC 2013
"Split horizon" for multicast payload frames

- Here: 47.31KBit/s vs. 14.73KBit/s
- Eliminates next bottleneck: ADSL upload
Distributed Address Table

- Distributed Hash Table for IPv6 Neighbor Discovery
- Like current DAT (Distributed ARP Table), but for IPv6, too
- Eliminates current largest ICMPv6 overhead:
  - IPv6 Neighbor Discovery
- Status: Patchset submitted, not upstream yet
Multicast Awareness

- Send multicast packets to interested nodes only
- Removes most ICMPv6 overhead:
  - Neighbor Solicitations, MLD Reports, ...
- Status: First basic patchset submitted
B.A.T.M.A.N. IV - Echo Location Protocol

- Perform link quality measurements with own packet type: ELP
- Reduced overhead through different intervals for OGMs and ELP
- Easier to optimize OGM propagation

Image Source: Petteri Aimonen, Wikimedia Commons, CC-BY-SA
Script-Kiddie: Mallory

- Playing with physical or virtual link layer
- Local disaster (mostly?)
Super Villain: Joker

- Knows batman well
- Global disaster
Current batman-adv algorithm is optimized for sparse networks
- Keep node neighbourhood small
Common LL-Service-Announcement protocols:
- Don’t scale... :(
With this flat and VPN architecture, batman-adv / layer 2 meshing works with:
- 80 nodes: without multicast filters
- 300+ nodes: with multicast filters
Eliminating Layer 2 specific overhead:
- Is on the horizon